Augmenting User Capabilities through an Adaptive Assistive Manipulator

Martin F. Stoelen

Directors:
Carlos Balaguer
Fabio Bonsignorio
Alberto Jardón Huete

Departamento de Ingeniería de Sistemas y Automática
Escuela Politécnica Superior
Universidad Carlos III de Madrid

Augmenting User Capabilities through an Adaptive Assistive Manipulator

Augmenting User Capabilities through an Adaptive Assistive Manipulator

Martin F. Stoelen

Directors:
Carlos Balaguer
Fabio Bonsignorio
Alberto Jardón Huete

Doctoral Thesis

Leganés, July 2014

Departamento de Ingeniería de Sistemas y Automática
Escuela Politécnica Superior
Universidad Carlos III de Madrid

Universidad Carlos III de Madrid

Publication Data:
Martin F. Stoelen
Augmenting User Capabilities through an Adaptive Assistive Manipulator
Universidad Carlos III de Madrid

ACKNOWLEDGMENTS

Dedicated to my wife Bea, my family, and my friends.

Special gratitude to my three supervisors for all their support, this would not have been possible without you. Thanks to Carlos and Alberto for introducing me to the field of assistive robotics, and for welcoming me at UC3M. Thanks to Fabio for introducing me to the world(s) of natural and artificial intelligence, and for priceless discussions and brainstorming.

I would like to also acknowledge my previous supervisors, Prof. David L. Akin at the University of Maryland, where this thesis begun, but also Prof. Christopher Welch, then at Kingston University. I am indebted to my current and past colleagues, and students. Thanks to Virginia in particular, for all your assistance during the last five years.

Finally, thanks to Prof. Angelo Cangelosi and the CRNS for great collaboration.

ABSTRACT

Assistive robot manipulators have the potential to increase the independence of disabled persons in activities of daily living. The current designs are mainly limited to pure teleoperation by the user, given the need for keeping the user in the control loop, and the complexity of the tasks and environments in which they operate. This thesis aims to augment the user's capabilities for performing such tasks by adapting the robot, and its level of assistance, to the user. Methodologies for modeling and benchmarking the complete human-robot system were established, which helped drive the development of different approaches to adaptation. This included a task-oriented optimization of the robot physical structure, approaches for low-level adaptive shared control, and work on interactive learning of, and assistance on completing, simple object manipulation tasks. Three experimental platforms were used: The ASIBOT manipulator of Universidad Carlos III de Madrid (UC3M), the AMOR manipulator of Exact Dynamics, and the iCub humanoid robot.

RESUMEN

Los manipuladores asistenciales tienen el potencial de incrementar la independencia de personas discapacitadas en sus actividades de la vida diaria. Los diseños actuales se limitan principalmente a una pura teleoperación, pues dada la complejidad de las tareas y del entorno, se necesita mantener al usuario en el lazo de control. Esta tesis pretende mejorar las capacidades del usuario para realizar estas tareas, adaptando el robot y su nivel de asistencia a las necesidades del usuario. Se han establecido metodologías para el modelado y evaluación del comportamiento del sistema formado por humano y robot, lo que ha permitido el desarrollo de diferentes aproximaciones a la adaptación. Esto incluye desde la optimización de la estructura del robot atendiendo a las tareas, la evaluación de diversas aproximaciones al control compartido adaptativo a bajo nivel, al aprendizaje interactivo y el desarrollo de asistencias para completar tareas sencillas de manipulación. Se ha hecho uso de tres plataformas experimentales: el manipulador ASIBOT de la Universidad Carlos III de Madrid (UC3M), el manipulador AMOR de Exact Dynamics y el humanoide iCub.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. BACKGROUND

1.1 Assistive Manipulators

1.1.1 Introduction

Most human manipulation tasks require the use of the upper limbs. Any deficiency in this part of the body causes loss of dexterity and a reduced performance in manipulation. These difficulties affect common Activities of Daily Living (ADL) such as moving objects, the use of tools or utensils, washing, dressing, opening and closing doors and drawers, turning switches on/off as well as eating and drinking. Assistive robots have the ability to aid in these tasks and can provide personalized assistance to persons with physical disabilities, individually or in teams (Mataric et al. 2007). They can supplement caretakers in helping, supporting, and monitoring persons with needs and can give the user increased daily independence. This thesis will focus on assistive robots that can assist the user in physical manipulation tasks, hereby denoted as assistive manipulators. These robotic technologies can be divided into three main development concepts: static systems that operate in structured environments, wheelchair-mounted systems, and mobile manipulator companions that follow the user around for personal and care applications.

1.1.2 Static Systems

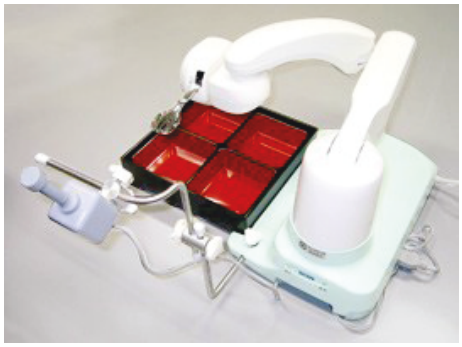
Static assistive manipulators are very useful when the user needs help in the same reduced living environment, always for the same applications. For example eating, drinking, washing, shaving, etc. They are low-cost in comparison to more advanced assistive manipulators, typically on the order of €3000-€5000. The Handy 1 robot arm (Topping 1993) is an excellent example of a static robot system, a 5 Degrees Of Freedom (DOF) manipulator aimed mainly at eating tasks, see Fig. 1.1a. The SECOM My Spoon (Fig 1.1c) is a more recent example of a 5 DOF feeder. It allows automatic feeding from predefined locations, but also manual control with a joystick to for example fine-tune the grasping of a given food item. A simpler system can be found in the Winsford Feeder (Fig. 1.1b). Here the plate is made to rotate, which means a simpler manipulator can be used. Another simplified system is the Sammons Preston Meal Buddy (Fig. 1.1d), which uses a 3 DOF manipulator to collect food from specially designed containers where little final adjustment is needed to collect the food. Research on such static "feeders" is ongoing, for example for specific food types (Song & Kim 2009) or for enabling usage by a larger set of user types (Guglielmelli et al. 2009). While providing an important service for a great number of disabled persons, these robot systems have limitations. Changing their location can be difficult, and may sometimes be nearly impossible. Especially if required in different rooms, or on different floors. They are also very limited in the tasks they can perform, due to the simple end-effectors used and to the low payload capacity.

1.1.3 Mobile Platforms

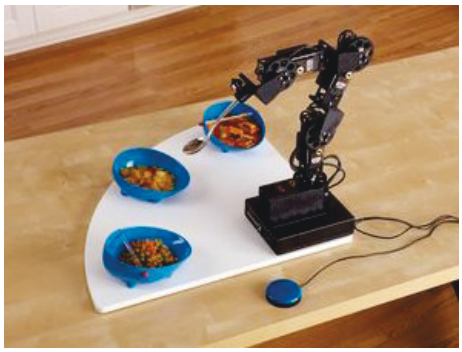
Another concept is the mobile assistive robot, which typically consists of one or more manipulators mounted on a wheel-based platform. While mobility around a domestic environment is not always ideal due to steps or obstacles, they are capable of moving independently of the

(a) Handy 1.

(b) Winsford Feeder.



(c) SECOM My Spoon.



(d) Sammons Preston Meal Buddy.

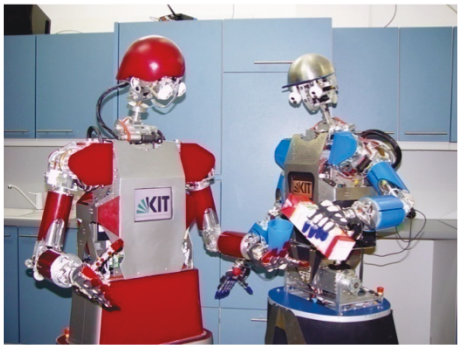
Fig. 1.1: *Example static manipulators, also known as "feeders".*

user in structured environments. Several examples of such systems exists, including the Korean KARES II mobile manipulator (Bien et al. 2004), the German Care-O-Bot (Graf et al. 2004) and ARMAR series robots (Asfour et al. 2008), as well as the American Willow Garage PR2 (Ciocarlie et al. 2012) and Cody (Chen et al. 2010). See Fig. 1.2. A characteristic of these platforms is the intended completeness in terms of actuation, sensing and autonomy. That is, they are typically aimed at being able to actuate based on high-level commands given by the user. The high complexity typically also make them expensive in comparison to other assistive robotic systems.

1.1.4 Wheelchair-Mounted Manipulators

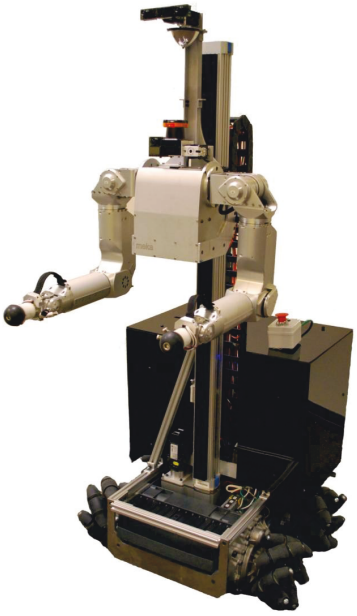
Manipulators mounted on wheelchairs is another class of physically assistive robots. These are also known as Wheelchair-Mounted Robot Arms (WMRA) (Kim 2012). The market leader of this type of robot has been the MANUS system of Exact Dynamics in Holland, now replaced by the newer model iArm. Other competitors include the RTD Raptor and the Kinova Jaco. See Fig. 1.3. Most of the systems commercially available have 5-6 DOF and weigh around 10 kg, and the price is typically on the order of €20000-€40000. For research purposes the AMOR of Exact Dynamics is interesting from the perspective of mobility, as it is a redundant manipulator with 7 DOF. Another example is ASIBOT, a 5 DOF manipulator with universal end-effectors that enables it to dock to specially designed docking stations in the environment (Jardon et al. 2006). Movement from one docking station to another is also possible, making it

(a) Care-O-Bot.



(b) ARMAR III.

(c) Willow Garage PR2.



(d) Cody.

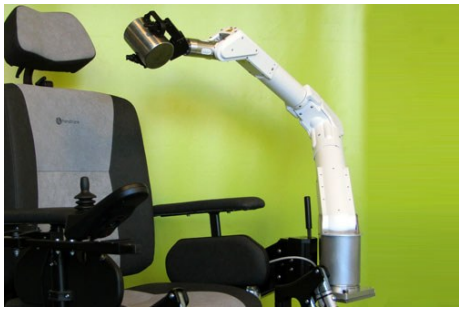
Fig. 1.2: *Example mobile assistive platforms.*

the only climbing assistive manipulator. This can help combat one limiting characteristic of such assistive manipulators, that the arm is permanently fixed to either the left or right-hand side of the wheelchair. This may become inconvenient for the execution of certain tasks. In addition it may produce mobility problems through stairs, doors, etc. The work presented here will mainly focus on wheelchair-mounted (and climbing) assistive manipulators.

1.1.5 Other Applications of Assistive Manipulators

While the main thrust of this thesis is to improve assistive manipulators for disabled persons, there are several very similar applications that may benefit from any novel technologies developed. This includes manipulators operating in close proximity with a human user with limited mobility, for example in Extra-Vehicular Activities (EVA) with joint robot and astronaut teams. Both on-orbit (Akin et al. 2003) and planetary (Fong & Nourbakhsh 2005). It also includes scenarios where a manipulator is directly teleoperated by a user, for example in space, underwater and disaster applications. Here the ability to effectively use the robot to perform tasks can be reduced

(a) ASIBOT.



(b) Exact Dynamics AMOR.

(c) RTD Raptor.

(d) Kinova Jaco.

Fig. 1.3: *Example assistive robot manipulators mounted to wheelchairs.*

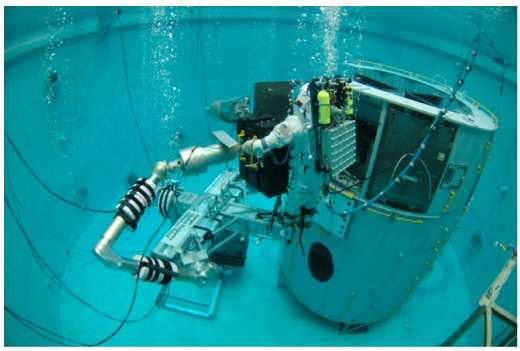
by several factors, for example time delays, limited camera views and dynamic environments. See Fig. 1.4a for an example of simulated on-orbit human-robot servicing of the Hubble Space Telescope. Part of the work leading up to the current thesis was focused on providing control over robots (and access to robot-related information) to astronauts working with manipulators on on-orbit servicing missions (Stolen et al. 2008). See Fig. 1.4b. Here automatic speech recognition, video and graphical displays were integrated on an experimental suit, the MX-2, and tested on realistic tasks in neutral buoyancy conditions in the Space Systems Laboratory at the University of Maryland, College Park. While the work helped outline some of the benefits of improved information flow to the astronaut, and the limitations of speech recognition in a noisy atmosphere, it was judged as out of scope for this thesis and will not be discussed further here.

Other related applications include arm prostheses and exoskeletons. See (Dellon & Matsuoka 2007) for an overview. As an example, the DEKA Arm seen in Fig. 1.5a is one of the most advanced upper limb prostheses in development. Recent work has also been performed on using muscle stimulation to help reduce the symptoms of tremor (Gallego et al. 2013). It is likely that the boundaries between such assistive technologies and the assistive manipulators described here will be gradually wiped out. This opens the exiting possibilities of applying some of the concepts developed in this thesis on for example arm prostheses, or applying for example muscle stimulation to help a user control an assistive manipulator. Medical robots used for surgery, like the Intuitive Surgery DaVinci system (Fig. 1.5b), may also be a potential application.

1.1.6 Open Challenges

Given the current state of the art of assistive manipulators, at least three open challenges can be identified:

1. *Approaching human-level performance on manipulation tasks.* This is difficult, given a user



(a) *Astronaut in MX-2 suit on the 6 DOF Ranger positioning leg, receiving assistance from two 8 DOF dexterous manipulators.*

(b) *Neutral buoyancy testing of interfaces for controlling, and receiving feedback from, robots in the MX-2 suit.*

Fig. 1.4: *Work on human-robot collaborative servicing of the Hubble Space Telescope at the Space Systems Laboratory, University of Maryland.*

(a) *The DEKA Arm upper limb prosthesis.*

(b) *The Intuitive Surgery DaVinci.*

Fig. 1.5: *Other potential applications of the work presented.*

with a low amount of mobility attempting to control a high-DOF robot safely and effectively in a partially structured, and potentially dynamic, environment. Current assistive manipulators typically move slowly to enable the user to maintain control over the arm. That is, faster actuation is feasible from a technical point of view, but limited mainly by two factors: a) a low information throughput between the user and the robot, and b) the challenges in making an assistive manipulator operate robustly and safely with autonomy. Any progress in these directions would have to be convincing enough to challenge the perception that assistive manipulators should necessarily be slow [1].

2. *Overcoming public opinion with respect to assistive robotics.* The European Commission in September 2012 published the results of an extensive survey on the attitude of the European population towards robots (Commision 2012). Based on 26,751 interviews, the survey revealed several interesting and very relevant trends for the thesis presented here. The principal being that 60% of the overall population felt that robots should be banned in the area of "care of children, elderly and the disabled", and 27% in the area of "healthcare". The corresponding numbers for where the interview subjects thought robots should be used as a priority were 4% and 22%, respectively. These results are likely a reflection of an (understandable) concern about the use of autonomous robots in these fields (see related

[1] Personal communication with a leading assistive technology company.

discussion in the literature on "robot nannies" (Sharkey & Sharkey 2010), (Belpaeme & Morse 2010)), but should also be taken into account when designing assistive manipulators that are directly controlled by the user.

3. *Maintaining assistive manipulators affordable to buy, simple to use and robust in operation with increasing system complexity.* That is, applying the right level of technology for the application at hand. A simple mechanical arm support may for example be a much more suitable aid than a full assistive manipulator for some users.

The core idea of this thesis is that these three challenges can best be approached by maintaining the disables user in the control loop, while enabling the assistive manipulator to use its own sensors to help and adapt to (and with) the user.

1.2 Human-Machine Adaptation

1.2.1 Introduction

Humans develop the ability to manipulate objects in the environment over the first few years of life, successfully overcoming the "curse of dimensionality" in spite of initially having very limited motor skills. The learning process is highly interactive, both with the environment and with human teachers. Such adaptation over the life of an individual is an important part of surviving in the open-ended environment we live. If an assistive manipulator is to operate in the same environment, some adaptation may also be beneficial. This could be adaptation to a given user's abilities or disabilities, but also to the task to perform and the typical scenarios in which the robot is used. At the same time the user is perfectly capable of adapting back, responding to changes in the system. This section will outline some of the principal currents in learning and adaptation, both artificial and natural, focusing principally on adaptation in the context of a human-machine interface. That is, adaptation in a system where a human and a machine need to interact to successfully perform a given set of tasks.

1.2.2 Inspiration from Cybernetics, AI and Cognitive Science

The period shortly after World War II saw large developments in among other computing machines, theories for learning and adaptation (in humans and machines), as well as human-machine interaction. Central to this progress were the many scientists and engineers that had been working together during the war on improving among other radars, guiding systems for anti-aircraft guns and cryptology. Several groups of such individuals formed, defining a field that was to be known as "cybernetics" (a name popularized by Norbert Wiener (Wiener 1965)). Among the British scientists the "Ratio Club"(Holland & Husbands 2011) was perhaps the most famous group, consisting of an interdisciplinary mix of biologists, mathematicians, engineers and psychologists, and including among other Alan Turing as member. William Ross Ashby was another, whose definition of adaptive behavior is well worth mentioning:

" ... "adaptive" behaviour is equivalent to the behaviour of a stable system, the region of the stability being the region of the phase-space in which all the essential variables lie within their normal limits." (Ashby 1960)

From this point of view, any adaptation should attempt to maintain the system from exceeding its limits for all variables that are important for the functioning of the system. Translating to a system comprising an assistive robot and its disabled user, to prevent damage or injury to the user, but also to make sure the user at no time during the interaction finds the robot too difficult

to use, or too slow in performing the desired tasks (leading to the user abandoning the robot). It should be noted that within such a system the user also has a great capacity for adaptation, a topic which will be treated further in Chapter 2. From cybernetics sprung several new fields, one of the most famous being Artificial Intelligence (AI), officially named at the Dartmouth Conference in 1956 (Kline 2011). One of the key early distinctions of this field was that of using symbolic manipulation in computers to attempt to develop machines that could think just like humans. That is, to move from the brain modeling of the early cybernetics to modeling the mind in a top-down way, starting with logical reasoning and working downwards to simpler activities.

From AI the subfield of machine learning is perhaps the most relevant for the work presented here. This subfield can be subdivided in many ways given the large variety of methods that fall under the umbrella of machine learning. One of the most common is to focus on the information available to the algorithm on the objective of the learning, with three main categories: supervised, unsupervised and reinforcement learning. The first, supervised learning, is typically used to denote approaches where the learning algorithm is given the exact desired output, and where the task is to learn a mapping from the input to its own output that maximizes a scoring function. For example by minimizing the error with respect to the desired output. Examples include learning based on back-propagation in Neural Network (NN)s, but also Gaussian Process Regression (GPR) and support vector machines (SVM). These are commonly applied also in robotics, for example in learning self-localization (Yamano et al. 2004), and object grasping (Pelossof et al. 2004). One important characteristic of many supervised learning methods is that they are non-local, i.e. each learning "unit" (e.g. an artificial neuron) require information from beyond its immediate context. For some algorithms this means that scaling to higher dimensions require exponential increases in computing power. It can also limit the ability of an algorithm to run online. It should be noted that supervised learning in AI overlaps with, and to a certain extent uses, mathematical optimization techniques. Indeed, in engineering optimization problems the goal is often to minimize errors between desired and actual outputs of a mathematical model by finding optimum sets of parameters. As an example, Chapter 3 of this thesis applies a heuristic optimization algorithm for finding sets of kinematic parameters for a given assistive manipulator that maximizes performance on a given set of tasks.

For scenarios where robots actuate in the real world, information on the exact desired output is often not available, and the robot may have to make do with sparser feedback from its environment. For example a discrete yes/no feedback on wether an action had a desired outcome, that is learning by trial and error or reinforcement. Reinforcement learning has been actively studied for robotics applications over the last 15 years, with several notable successes. For example learning to successfully flip pancakes (Kormushev et al. 2010) and playing ball-in-a-cup (Kober et al. 2010) with high-DOF manipulators. However, reinforcement learning approaches often run into the "curse of dimensionality" when trying to relate the state and action spaces for high-dimensional real-world problems. Most approaches also assume that the robot has access to the exact state, or a set of sensor values that accurately describes the state. And for applications with humans interacting with the robot, any approach should keep the number of trial-and-error attempts low, at least to the tens or hundreds. This can be an issue in complex problems, given the sparse information, often a single scalar, received through the reinforcement signal. There are also learning approaches that aim to learn without explicit information being provided on goals or performance, so called unsupervised learning. Here the goal is typically to discover relationships and correlations in the input provided, for example by performing clustering on the data. Methods performing unsupervised learning include Self-Organizing Maps (SOM) and Slow Feature Analysis (SFA) (Wiskott & Sejnowski 2002). In robotics unsupervised learning has been used to for example estimate the traversability of outdoor terrains (Kim et al. 2006).

Although machine learning approaches have had impressive successes, traditional AI has

been criticised for its (mainly) symbolic approach to endowing machines with intelligence. While symbolic manipulation is something computers excel at, it is an open discussion wether it is what is really going on in the biological brain and wether a computer can be used to perform as a biological brain. See for example (Searle 2004). Using symbols to create robust actuation in real world applications can be difficult, given that they have to be used in conjunction with a large amount of noisy sensor input to generate sub-symbolic commands to the actuators of the robot. This is related to the symbol grounding problem (Harnad 1990), which is further discussed in Chapter 6. Another critique is that traditional AI research typically does not take advantage of the physical embodiment of the agent, and its interaction with its environment. That is, some of the computation required to control an embodied agent can be offloaded to the physical implementation, through for example designing for passive stability and adaptation (Pfeifer et al. 2007). This is also known as morphological computation, and can be found extensively applied in nature. For example in the extremely efficient (both in terms of energy and control requirements) pendulum-like walking gait of humans, replicated in passively walking biped robots (Collins et al. 2005). Using embodied sensing is further explored in the later chapters of this thesis.

It has also been pointed out that a brain embodied in a biological system develops mental capabilities through *"autonomous real-time interactions with its environments (including its own internal environment and components) by using its own sensors and effectors"* (Weng et al. 2001). This type of development has been extensively studied in psychology and neuroscience. Recent years has also seen an increasing mass of work trying to allow robots to follow such a developmental process (Asada et al. 2001), also known as developmental learning or autonomous mental development. For example the automatic learning and organization of skills for interacting with the environment. See for example (Yamashita & Tani 2008). Chapter 6 describes a developmental approach to learning object manipulation skills in real-time with a human teacher. Finally, it is worth mentioning that both traditional AI and much of contemporary neuroscience assume a set of specialized centers or modules for performing perception, planning, and actuation etc. This is also related to the view of memory as a storehouse, for example assuming that the brain stores visual images of previously seen letters for letter recognition, which has been criticized for not being able to explain the ability of biological agents to relate past experience, the present sensory context, and the resulting actuation:

"However, the many ways in which similar motor acts can be performed (with the right hand, the left hand, the foot, and so on), as well as the overriding importance of context in recognition, suggest that neither localization nor modularity adequately explains the nature of brain function." (Rosenfield 1988)

As shown above, the ongoing discussion on how to approach natural and artificial intelligence (and learning) can be both stimulating due to the diversity of opinion, and frustrating given the lack of consensus on how to best proceed. The stance taken in this thesis is to carefully evaluate the requirements of the adaptation for assistive manipulators, and to choose learning methods that can fulfill these requirements, regardless of field. That is, to attempt to take advantage of the width of work available on adaptation in cybernetics, in AI, and in cognitive science.

1.2.3 Adaptation in the Human Factors Literature

Adaptation in automation has been shown to reduce some of the problems with the interaction with automated systems, such as unbalanced mental workload, reduced situational awareness and complacency by the user. See (Hancock et al. 2013) for a representative overview of human-automation interaction research in the Human Factors literature. However, adaptation by the system can also have a negative influence and is somewhat controversial for human-machine

interfaces in general. Work has for example been done on adaptive graphical interface design, for example adjusting the ordering of menus based on the frequency of usage of each menu item. While some studies have found improvements with this type of dynamic adaptation, others have found a reduction in performance (Mitchell & Shneiderman 1989), with the suggestion that the user should be able to decide when the adaptation should occur (Findlater & McGrenere 2004). One reason why adaptation is a difficult topic in interaction with a user can be deduced from the 8 golden rules of interface design (Shneiderman & Plaisant 2004). If the interface presented to the user, or similarly the functionality of a system, changes from one use to another (if for example it attempts to adapt to the user), then this challenges both the need to *"strive for consistency"* and to *"support internal locus of control"* in the interface design. That is, the adaptation might frustrate the user if the desired functionality is no longer available from the same user input, and the user may feel that this loss of predictability makes the system more difficult to control to achieve her/his goal. Not to mention the possibility of the adaptation actually worsening long-term performance.

1.2.4 Adaptation in Assistive Technologies

(Vanacker et al. 2006) presented an adaptive strategy for filtering the commands coming from the disabled user of a wheelchair using contextual information from sensor readings as well as previous data from able-bodied users on the same tasks. A layered feedforward NN with self-recurrency in each hidden layer was used, which were trained offline and used online. The results showed improvement in the time, velocity and errors of a user in a simulated environment. The testing was performed with one of three simulated disabilities added to the user commands during operation. All three disabilities were of a deterministic nature. A more recent approach used plan recognition to obtain the probability of a set of user plans, given a set of observed user commands (Demeester et al. 2008). Each plan is a trajectory of translational and rotational velocities in the plane from the current state to a given goal in the environment. The goals can be given beforehand or obtained through observation of the user's behavior over time. Another approach for predicting the intent of a wheelchair user was shown in (Carlson & Demiris 2008), where specific local models for actions, for example moving towards a door, is used. A confidence function, including for example Euclidian distance and heading angle, is used to estimate the confidence that the user wants to perform an action. If the confidence is above a given threshold, assistance is given by the chair directing the movement, while the user controls the speed along the trajectory. Such systems have been shown to reduce both collisions and cognitive workload (Carlson & Demiris 2012). Learning has also been applied to assistive robot walkers. See for example (Patel et al. 2010), where a Dynamic Bayesian Network (DBN) learns a subset of possible actions of the user, and then adapts the behavior of the walker to the user's current action. There is also work on adapting the social interaction between an assistive robot and its users in post-stroke rehabilitation (Tapus et al. 2008).

1.2.5 Key Design Drivers

Three key design drivers were extracted from the above survey on adaptation and human-machine interaction:

1. *Enable online incremental adaptation in the robot.* Given the potential for changes in the system to make the interaction frustrating for the user, it is here postulated that any adaptation should be as smooth as possible. That is, small incremental changes are preferable to, for example, adaptation that only occurs between each usage of the robot. This allows the user to immediately notice and respond to changes that are undesirable,

in effect "adapting back". The role of such a mutual adaptation in creating a symbiotic relationship between user and robot is one of the central questions in this thesis. One consequence of the need for incremental adaptation is that all learning algorithms should run online, and in real-time with the user. This will heavily influence the developments in the later chapters of this thesis, where an adaptive shared control between user and robot is intended.

2. *Provide feedback to the user on the robot state and intent.* Related to the above point, another design driver of importance is to provide the user with rich information on the robot. This should include information on the robot's perceived state with respect to the world. For example auditory feedback on proximity to obstacles, see Chapters 4 and 5. It may also include ways for the user to understand the intent of the robot when performing tasks, for example through visualizations of the planned trajectories. Again the idea is to make the actions performed by the robot as predictable as possible from the user's perspective. Feedback on predicted user movements can also help the user asses the need for additional learning in the robot. This is further explored in Chapter 6.

3. *Approach the interaction as a closed-loop information transfer.* It is postulated that both the explicit (i.e. joystick commands) and implicit (i.e. common detection of an obstacle) interaction occurring between user and assistive manipulator can be modeled as a two-way transfer of information. This transfer can occur through the environment, for example when the user is observing the effect of the robot actions on objects in its surroundings. Thus interaction is here seen as any information that can pass from the user to the robot, and back again, through any available medium. Information Theory (Shannon 1948) can then also be used to quantify the interaction, which will be further discussed in Chapter 2.

1.3 Scope of Thesis

The scope of the thesis is here defined, inspired by the identified open challenges for assistive manipulators, and limited by the key design drivers for adaptation in human-machine interaction. First of all, it is clear that methods from artificial intelligence will be required, but the application will here be limited to situations where the user is in the control loop (i.e. focus on non-autonomous operation). Second, the developments of the thesis aim to provide physical assistance through assistive manipulators, and the thesis will therefore not include social robots and their potential cognitive assistance. Third, the thesis is aimed at physically disabled but cognitively able users. That is, users that have no reduction in cognitive abilities, for example in understanding how to open a door, even though they may need assistance in physically performing such a task. Finally, the methods developed aim to be as general as possible, i.e. suitable for a range of disabilities and tasks.

1.4 Organization of Thesis

The thesis is organized into six main chapters in addition to the current chapter and the conclusions. Chapter 2 will focus on modeling the complete system and its most important subparts. This also includes investigation into how to benchmark assistive manipulators and what metrics to use. These are used for quantifying changes in performance in some of the later chapters of the thesis. Then follows four chapters implementing approaches for adapting an assistive manipulator to the user, based on the analysis in Chapter 2. Finally, Chapter 7 attempts to put together the different approaches and proposes an architecture for helping the user of an assistive manipulator complete his/her daily tasks, achieved through an architecture

with adaptation on several levels. While the thesis covers a broad range of topics, it is based on a small set of common concepts. These are shown in Fig. 1.6, which also indicates the usage of the concepts in the six main chapters.

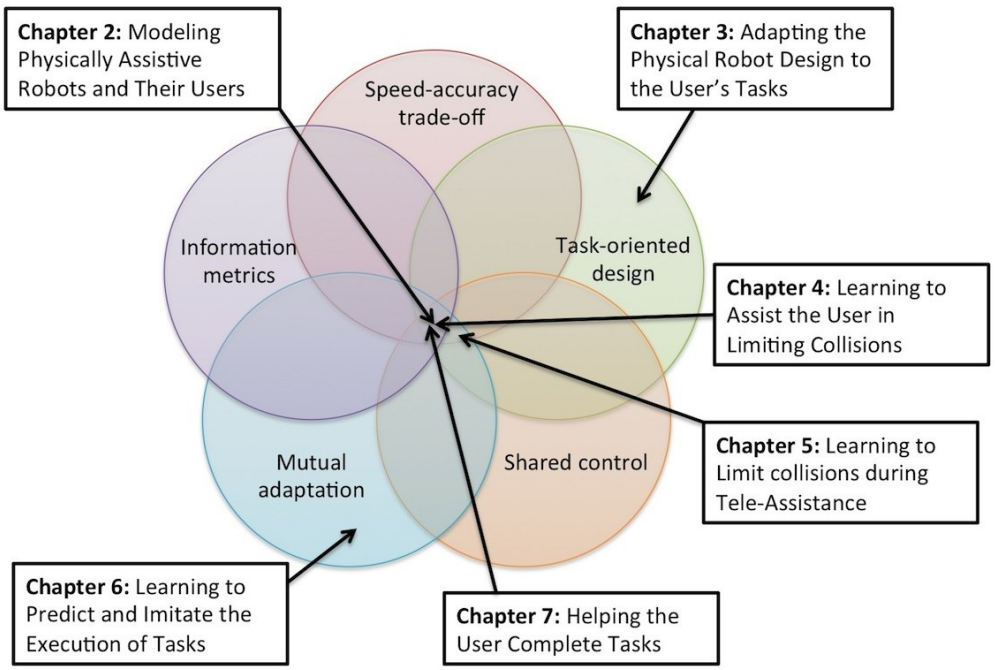


Fig. 1.6: *Diagram of five key concepts involved in the thesis, and the use of these concepts in the six main chapters. The speed-accuracy trade-off refers to the inherent limitation of human movement generation, where moving faster typically means moving less accurately. Task-oriented design of robots attempts to quantitatively take into account real tasks and obstacles in the design process, typically through a numerical optimization of the physical structure. The shared control concept is key to the work presented here, and means that both the user and the robot itself can influence the execution of tasks. If both the robot and the user can adapt, such a closely coupled system can include a mutual adaptation between the two actors. Finally, metrics based on Information Theory can give new insights, and measure the performance of, complex sensorimotor systems like the ones explored here.*

2. MODELING PHYSICALLY ASSISTIVE ROBOTS AND THEIR USERS

2.1 Introduction

Good methodologies for modeling and benchmarking can help drive the development of robots in general, and assistive manipulators in particular, given the complexity of the human-robot system involved. By modeling is here implied understanding what components constitute the system as a whole, their principal characteristics, and their interrelationships. Section 2.2 will investigate system-level models of assistive manipulators (including their users) with an emphasis on the interaction and transmission of useful information. Section 2.3 will address one of the principal characteristics of the human capability for generating movements, the trade-off between speed and accuracy. The modeling and simulation of common disabilities are explored in Section 2.4. By benchmarking is here implied quantifying the performance of the system as a whole and the strength of the interrelationship of its components, in a way that can be compared across conditions and system configurations. See Section 2.5 for a discussion on benchmarking and scientific method in assistive robotics.

2.2 Modeling the Complete System

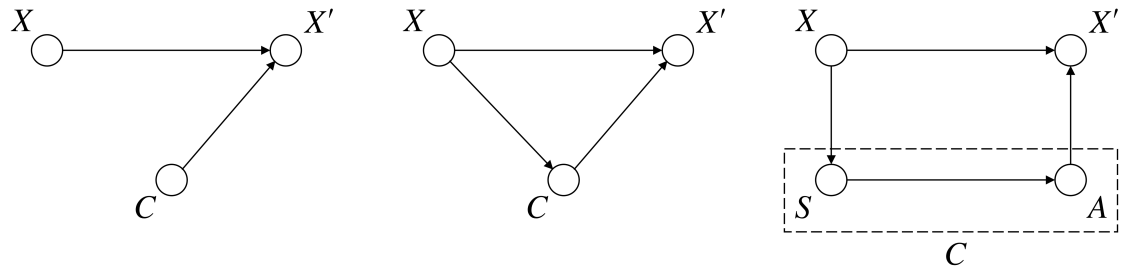
2.2.1 Definition of the Problem

A model of the complete system for an assistive manipulator will typically have to include the user, an adaptive and self-motivated agent, inside the control loop. Mathematically modeling the user is hard, typically only feasible for specific actions or tasks, while a large set of diverse tasks are required. The inputs and outputs of the overall system are not easily defined either. The same model should be useful for the range of different input modalities that the intended users require, so dealing with different representations of the interaction, from low-level joystick deflections to high-level speech commands. The sensor data can also vary, given the requirements of the tasks the user wants to perform. Modeling is also complicated by the fact that human-systems interactions can often exhibit dynamic nonlinear properties and chaotic behaviors (Karwowski 2012). It is here therefore argued that what is really of importance is the information that is transmitted in the system during usage, and its usefulness in accomplishing the user's goals in an effective and safe way.

2.2.2 Information Theoretic Models of Control Systems

(Touchette & Lloyd 2004) proposed an information-theoretic approach to the study of control systems, with the aim of defining a methodological framework for the control of chaotic maps and of stochastic non-linear dynamical systems in general. This approach is based on representing a complete control system as a directed acyclic graph of random variables, see Figure 2.1, and analyzing it using concepts from Information Theory (Shannon 1948). S. Lloyd is a pioneer of quantum computation and quantum communications, in particular he proved quantum analogs

of Shannon's noisy channel theorem, and introduced novel methods for quantum error correction and noise reduction. The system model includes the current state X, with values $x \in \mathcal{X}$, and the future state X'. The random variable representing the controller, C, then senses the current state (with sensor S) and actuates to achieve the future state (with actuator A) . This can be represented by conditional probabilities, $p(c|x)$ and $p(x'|x, c)$. These can be viewed as representing a sensor and actuation channel, respectively. The authors were further able to derive the conditions for observability, controllability, and optimality using this method.



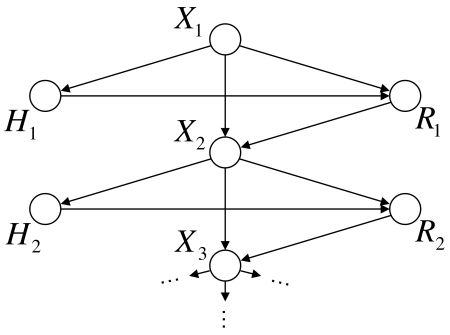
(a) *Open-loop control.* ***(b)*** *Closed-loop control.* ***(c)*** *Sensor (*S*) and actuator (*A*) constituting the controller (*C*).*

Fig. 2.1: *A control system as a directed acyclic graph (Touchette & Lloyd 2004).*

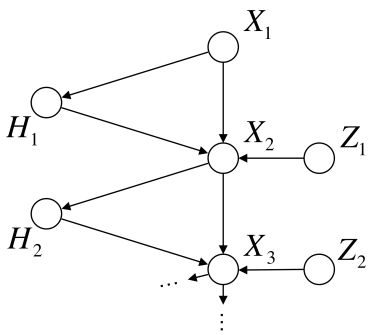
The work in (Touchette & Lloyd 2004) built on previous work by S. Lloyd and Slotine, see (Lloyd & Slotine 1996), while the particular idea of embedding classical noisy channels into non-noisy channels, that is inherited here, is inspired by a widely used formal method exploited in quantum information theory, see (Bennett & Shor 1998). An entropy approach to the problems of the control of stochastic non-linear systems was previously proposed (with different mathematical formalisms) by Saridis (Valavanis & Saridis 1988), (Saridis 1988), (Saridis 1995). Recently, (Liu et al. 2011), interesting methods for the quantitative characterization of the control of complex networks have been proposed. The advantage of the methods proposed in (Touchette & Lloyd 2004) with respect to those proposed by Saridis are that, while sharing the conceptual framework, they lead to a more direct quantification. The models in (Liu et al. 2011) are on the one hand too complex for the application considered here, on the other hand they do not provide, directly, quantitative measures for learning processes. Approaches using Information Theory have been suggested in the field of Human-Robot Interaction (Gold 2009). Information metrics are also employed to understand networked embodied cognition, for example the relationship of robot morphology and computation (Bonsignorio 2007).

2.2.3 Modeling Complete Human-Robot Systems

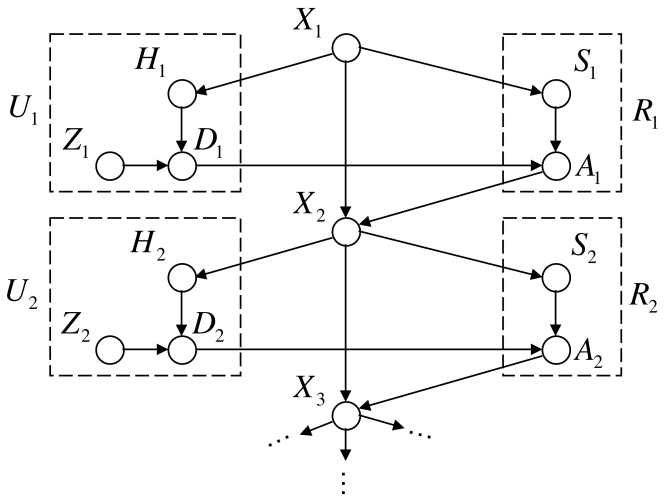
A representation of the principal interaction between a user and an assistive robot in general is shown in Fig. 2.2a. For each time step t the system has a random variable representing the current state X_t, with values $x_t \in \mathcal{X}_t$, and a future state X_{t+1}. Time is therefore in the vertical direction here, with three time-steps of the state shown for clarity. The random variable representing the human intent, H_t, depends probabilistically on the current state. Thus the human is able to observe the robot in the environment directly. At the same time the robot R obtains its own belief about the current state of the system and actuates based on this information and the input from the user to generate the future state X_{t+1}. The state of the system is here assumed to include the physical robot, the physical human body and the physical objects and obstacles relevant to the task.



(a) *Model of the interaction between the human and the assistive robot.*



(b) *Model of the disabled human's perspective on robot control.*



(c) *Model of a disabled user operating an assistive manipulator with sensors S_t and actuators A_t.*

Fig. 2.2: *Directed acyclic graphs representing simplified versions of the complete system, including user and assistive robot in the environment. Subscripts indicate time-steps.*

Fig. 2.2c shows the model for a disabled user attempting to control an assistive manipulator. A random variable for the user, U_t, here includes the random variable representing the human intent, H_t, which the user intends to actuate over an input device, D_t. Note that multiple input devices could potentially be used simultaneously, but is omitted for clarity. As can be seen from the Figure, this random variable depends probabilistically on both the user's intent H_t and a noise Z_t. The noise is explicitly included to represent the part of the interaction that does not depend statistically on the intended message of the human acting as the source, and is here assumed to represent a generic physical disability. The robot is expanded. The random variable S_t is here assumed to represent the robot's interpretation of the sensed state. The robot actuation, A_t, is assumed to represent the robot's interpretation of the action to be executed, based on the sensed state and the commands received from the user. See Section 2.2.5 for a description of Fig. 2.3.

2.2.4 Quantifying Correlations in the System

Given the models in Fig. 2.2, the conditional probabilities representing the correlations of the different random variables can be defined. For example, the probabilistic dependencies of the actuation of the robot for the specific model in Fig. 2.3 can be written as $prob(a_t|d_t, c_t^1, \cdots, c_t^n, p_t^1, \cdots, p_t^m)$. These correlations can be thought of as noisy communication channels between the random variables. The information in a correlation can be represented formally by the mutual information (Cover & Thomas 2012), introduced by Shannon (Shannon 1948). See Equation (2.1) for the case of two random variables. The required probability distributions can be estimated, for example by a sampling process that counts how often a given value of one or more random variables occur, and metrics based on mutual information can therefore be calculated from experimental data. See for example (Lungarella et al. 2005), (Bonsignorio 2007), (Bonsignorio 2013).

$$I(X;Y) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x,y) \log_2 \frac{p(x,y)}{p(x)p(y)}. \tag{2.1}$$

The overall goal of the system design can then be phrased and potentially also quantified experimentally; to maximize the mutual information between the user's intent and the robot actuation, i.e. maximize $I(H;A)$. For most tasks however, in experimental settings and in real life usage, the user's intent is not easily predicted (unless it is controlled for). A more readily available information may be the noise Z, which can for example be a simulated disability in an experimental setting. See section 2.4.4.

2.2.5 Metric: Controllability from the User's Perspective

Controllability is a useful metric for controllers acting in a closed-loop in general. Loosely speaking, a system is controllable if it can be commanded to any final state from any initial state with a probability of one. Here we show that controllability is also relevant for the specific case where the controller is a disabled user, H, trying to overcome the noise produced by his/her disabilities, Z, to successfully perform a manipulation task with a physically assistive robot. That is, a system like the one system shown in Fig. 2.2b, for which the necessary and sufficient conditions for controllability was defined in (Touchette & Lloyd 2004), and is outlined below. Under the assumption that the future state, X_{t+1}, is a deterministic random variable when conditioned on the values x_t, h_t, and z_t, this system can be said to be completely and perfectly controllable over the support of X if, and only if,

$$\min_{prob(h_t|x_t)} I(X_{t+1}; Z_t|X_t, H_t) = 0, \tag{2.2}$$

and $prob(x_{t+1}|x_t) \neq 0$ for all x_{t+1}. For the experimental evaluations performed in Chapter 4, the controllability is approximated using the unconditional mutual information over the robot actuation and the noise added, $I(A_t; Z_t)$, measured over a finite set of tasks and a fixed number of repetitions for each task. Loosely speaking, the controllability metric thus measures the influence of the noise added on the actuation of the robot. The user is assumed to attempt movements that are repeatable (i.e. passing through a similar trajectory of values for X_t and H_t) and close to optimal (i.e. the minimization over $prob(h_t|x_t)$) for each set of attempts used to calculate the controllability. The ability to reach any given state with a non-zero probability is also assumed.

2.2.6 Metric: Predictability of Execution

The above controllability measure can be complemented by other measures. A high value of the controllability approximation used can for example be achieved by providing randomized user input. That is, by violating the assumption that the user input is close to optimal for the task. (Bialek et al. 2001) proposed predictive information, in the form of the mutual information between the past and the future, as a general measure of complexity of a time series. The measure can be said to quantify the total information of past experience that can be used for predicting future events, and has among other been applied to the behavior of mobile robots in an unknown environment, see (Ay et al. 2008). Applying this measure to the user input may help identify cases where the controllability measure is no longer valid. Applied to the robot actuation it may be used to compare how "predictable" the actuation is with different control approaches. The one-step mutual information is here used, with the predictive information of actuation thus being defined as $I(A_t; A_{t+1})$. See Chapter 4.

2.2.7 Discussion

The above information-theoretic model is quite generic, as it does not specify what type of robot actuation is possible, what type of robot sensorial information is used, nor the specifics of the explicit interfaces over which the user can communicate with the robot, and vice versa. However, it does therefore also represent a large set of possible configurations, for example a 5 DOF assistive manipulator where the user commands the end-effector velocities (in the end-effector local frame) using a joystick, and where the robot uses this information together with distributed proximity and collision sensors to perform its tasks. This is the case shown in Fig. 2.3, the configuration used in Chapter 4. The robot has here access to n collision sensors C and m proximity sensors P. Another benefit of the model is the definition of quantitative metrics, which can be used to understand component and system-level behaviors, as well as measure performance. See Chapter 4 for an example application of the metrics defined.

Information-theoretic concepts have been used to identify various aspects of the structure of the sensorimotor space, whether by imposing an information limitation on models that an agent can form of its environment (Klyubin et al. 2007), (Moller & Polani 2008), or by embodiment- and dynamics-mediated models for self-motivated behaviour generation. For example by predictive information or empowerment, the external channel capacity of the perception-action loop of an agent (Ay et al. 2008), (Zahedi et al. 2010), (Klyubin et al. 2008). The information-theoretic view allows for a coherent, model-independent treatment of the quantities of interest, as they are all expressed as bits. The related work presented here can be seen as part of a framework

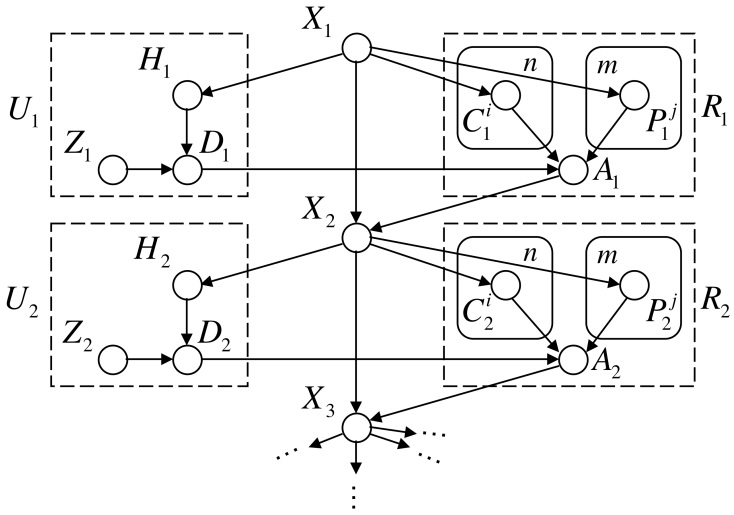
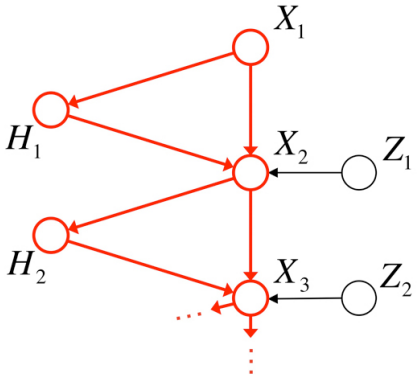


Fig. 2.3: *Specific model of the system in Chapter 4.*

for the quantification of the development of embodied cognitive networks (Bonsignorio 2013). An important aspect of the model used is that much of the interaction between user and robot occurs through the environment, included in the state of the system X. It should be noted that this is not a new idea for cognitive systems in general, as pointed out by Ashby:

"The anatomist may be excused for thinking that communication between part and part in the brain can take place only through some anatomically or histologically demonstrable tract or fibres. The student of function will, however, be aware that channels are also possible through the environment. An elementary example occurs when the brain monitors the acts of the vocal cords by a feedback that passes, partly at least, through the air before reaching the brain." (Ashby 1960)

2.3 Modeling the Speed-Accuracy Trade-Off of Human Movements



2.3.1 Introduction

Most targeted human movements are governed by a trade-off between speed and accuracy. That is, a reaching task can be performed with sufficient accuracy only up to a given speed of execution. Or conversely, maintaining a given velocity in a car can only be safely achieved if the road is of a sufficient width. The same speed-accuracy trade-off is therefore also present when a user is operating a human-machine interface. For example when controlling a robot manipulator. Such movements are high-dimensional, and in general require the user to control up to 3 DOF rotational movements and up to 3 DOF translational movements of the robot end-effector simultaneously, indeed like many movements of the human hand. Most assistive manipulators are controlled with 2 DOF joysticks, with switching between DOF required, but the task requirements are the same.

Many different user interfaces exist for this purpose. These include free-flying input devices that directly relate the rotational and translational pose of the input device to the object controlled. Other devices like joysticks are rate-based, where the deflection or force applied to the device is used to control the velocity of movement. However, determining the most suitable input device for a given 6 DOF application can be difficult due to the many variables involved (Bowman et al. 2004). One approach is to test each device in very specific scenarios, for example peg-in-the-hole tasks. This may be sufficient if the tasks are limited and well known, but for most applications the actual usage of an input device can involve any number of combinations of the translations and rotations available. Each could also have a very specific amplitude and accuracy requirement that needs to be coordinated. This makes it difficult to come up with a representative set of tasks for a comparison of these input devices. A model that could relate the relevant properties of a 6 DOF task with the completion time could help practitioners generalize to other similar tasks. This would make comparisons of these input devices less ambiguous and may help resolve lingering issues about their application.

One example is the Remote Manipulator System (RMS) flown on the Space Shuttle, which is controlled using one hand controller for rotations and one for translations. The benefit of separating the DOF between hands was promoted by (Hartley et al. 1986) to help reduce piloting errors arising in high-workload situations. One example given was that a single 6 DOF controller induced unwanted coupling between the controlled axes, such as unwanted roll or yaw rotations accompanying an intentional side-to-side translation command. However (O'Hara 1987) found little difference in errors for similar tasks, and some have argued that one 6 DOF controller is preferable (McKinnon & King 1988). One question here is the use of one versus two hands. Another is that of coordinating the rotations and translations required for the task. The studies primarily assessed completion time and errors for specific tasks like spacecraft docking, a typical approach for studies on high-DOF movements. Other measures have been proposed to quantify the degree of coordination (Zhai & Milgram 1998). However a model that can relate task performance with the distance and tolerance parameters of the task would also implicitly describe the coordination of rotations of translations.

Such a model could also be applied to the control of assistive manipulators, potentially driving the human-machine interface design.

2.3.2 Fitts' Law

Since its original publication, Fitts' law (Fitts 1954) has been an important tool in modeling the speed-accuracy trade-off in simple human movements. As seen in Equation (2.3), the model predicts that the Mean Time (MT) to complete a movement varies linearly with the Index of Difficulty (ID). This index is a function of the distance moved and the accuracy requirement, or tolerance, on the movement. These are denoted as the amplitude of movement, A, and the

width of the target area, W, respectively. Typically the application of the law is in simple left to right movements of the hand, here denoted as translational movements with one DOF. See Fig. 2.4 for the original task used by Fitts. The term translation is used to indicate the lack of a rotational requirement on the movement itself, although rotations about one or more joints will be necessary to perform the task. The coefficients a and b are determined experimentally using a linear regression analysis. The slope coefficient b then becomes a measure of the rate of change of completion time with change in the difficulty of the task and the reciprocal, $1/b$, is known as the Index of Performance (IP). In other words, human performance for a task with a given distance and accuracy requirement can be predicted on the basis of observations of other such combinations.

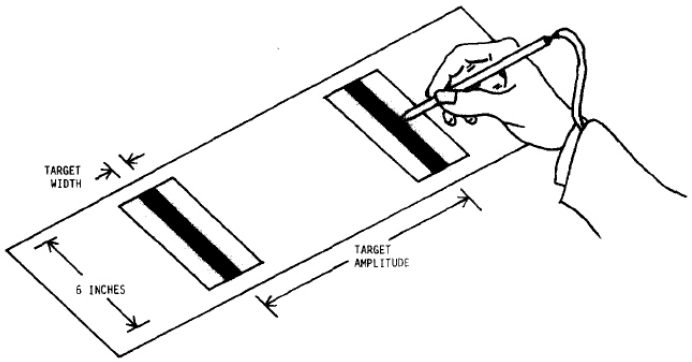


Fig. 2.4: *The original reciprocal tapping task used by Fitts (Fitts 1954).*

$$\begin{aligned} MT &= a + b \cdot ID \\ ID_{Original} &= log_2 \left(\frac{2A}{W} \right) \end{aligned} \tag{2.3}$$

The form of Fitts' law used here was first proposed by (Mackenzie 1989) and is shown in Equation (2.4). This version, which is also known as the Shannon formulation, has a better correspondence with the underlying Information Theory basis of the law, and has been shown to provide a better fit to experimental data than the one originally presented by (Fitts 1954). It is also the basis for performance testing in ISO 9241-9 (ISO 2000), which covers ergonomic requirements for non-keyboard computer input devices.

$$\begin{aligned} MT &= a + b \cdot ID \\ ID_{Shannon} &= log_2 \left(\frac{A}{W} + 1 \right) \end{aligned} \tag{2.4}$$

This standard also includes a recommendation for performing the adjustment for accuracy. This implies the calculation of an effective target width for each participant and each condition, based on the standard deviation of endpoints found during testing. An effective amplitude equivalent can be calculated from the actual distance moved by each participant for each condition. Substituted for W and A in Equation (2.4), an effective Index of Difficulty (IDe) can then be defined for each case. This should more closely represent the actual difficulty of the task for each participant and can be used to define a new measure, namely throughput (TP). Although TP is sometimes used to denote IP, the definition of TP used here follows that in (Soukoreff & MacKenzie 2004), as seen in Equation (2.5). This form is also known as the mean of means TP, a notation that will be used in the rest of this thesis. Although similar to the index of performance described above, it is considered to give a better representation of the actual performance of

the participants as it combines the speed and accuracy of the movement performance into one dependent measure.

$$TP = \frac{1}{y}\sum_{i=1}^{y}\left(\frac{1}{x}\sum_{j=1}^{x}\frac{IDe_{ij}}{MT_{ij}}\right) \tag{2.5}$$

Fitts' law has been used extensively in the field of Human-Computer Interaction (HCI) to quantify performance and drive graphical user interface design. The law is also commonly used in comparisons of input devices, where it provides the capability to generalize about results beyond a specific task. One of the early applications of Fitts' law in HCI was in the favorable comparative evaluation of the mouse with other input devices (Card et al. 1978). In fact, Fitts' law remains one of very few hard quantitative tools available to designers of human-machine interfaces, even though it is today considered more as an empirical regularity than as a model of the underlying mechanics of human movement.

Multidimensional versions of the law have been developed, including pointing in 2D and 3D (Grossman & Balakrishnan 2004). Rotational movements based on Fitts' law tasks have also been explored. Early studies followed up on Fitts' original study to determine the best representation of the difficulty of a rotary task (Knight & Dagnall 1967). It was found that the index of performance was similar to those found in translational movements (Crossman & Goodeve 1983), and that Fitts' law could represent rotational tasks reasonably well. Indeed the law has been extended to represent elbow flexion-extension (Kondraske 1994) and more recently been proposed for more complex models of human upper limb performance (Yang et al. 2001). The steering law (Accot & Zhai 1997) is an extension of Fitts' law to corridor-following tasks.

2.3.3 Signal-dependent Noise and Motor Planning

The intention behind Fitts' original experiment was to establish the information capacity of the human motor system. This was inspired by the information-theoretic approaches popular at the time, and more specifically the effect of noise in limiting the information capacity of a communications channel (for a detailed description see (MacKenzie 1992)). Trying to explain the empirical regularity inherent in the law using theories of human movement has since been an active research field, with a definite conclusion yet to be made. This includes the iterative-corrections model of (Crossman & Goodeve 1983), the stochastic optimized-submovement model of (Meyer et al. 1988) as well as more recent neurodynamic approaches (Beamish et al. 2006).

One prominent explanation for the speed-accuracy trade-off in Fitts' law is worth exploring here in more depth, namely the minimum variance model of (Harris & Wolpert 1998). This model postulates that a signal-dependent noise exists in neural transmissions, and that the motor planning attempts to optimize the final positional variance of the movement for a given movement time, or similarly the movement time for a given accuracy requirement, in the presence of this noise. The resulting movements have a bell-shaped velocity curve, see Fig. 2.5, which are also predicted by the model. The main property of the signal-dependent noise assumed for the neural control signal is that the strength of the noise is proportional to the mean level of the signal produced. Thus while moving quicker will reduce the time to move a given distance, it also increases the level of neural firing, the noise, and thus the variability of the movement (reducing its accuracy). This could for example translate into the inability to consistently hit within a given target in a Fitts' law type task.

It is interesting to see the above signal-dependent noise in the context of the Information Theory roots of Fitts' law, as well as in the context of the information-theoretic models of the human-robot system presented in Section 2.2.2. In fact, an artificial signal-dependent noise is used as a simulated physical disability in parts of this thesis, see Section 2.4.4 for details. In

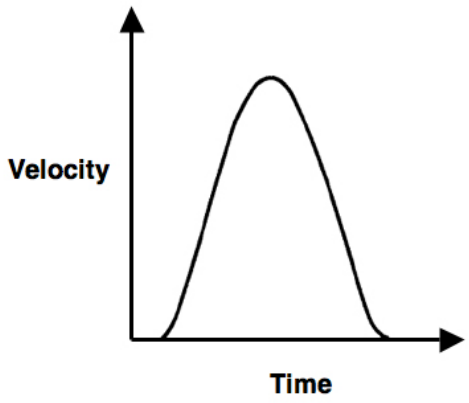


Fig. 2.5: *Example bell-shaped velocity profile.*

addition, the speed-accuracy trade-off caused by the noise is one of the inspirations for developing the adaptive collision-limitation behavior of Chapter 4. That is, the behavior can be thought to increase the tolerance required for a robot grasping task (from the user's perspective), through local distributed sensing. See (Faisal et al. 2008) for an overview of noise in the nervous system.

2.3.4 Extending Fitts' Law to Combined Movements

As outlined above, the previous applications of Fitts' law have focused on 1, 2 or 3 DOF translational tasks or 1 DOF rotational tasks. This section describes a study on extending the speed-accuracy relationship represented by Fitts' law to more complex movements involving both rotations and translations. Movements with 1 DOF rotation and 1 DOF translation were chosen. Separate 1 DOF rotations and translations have previously been modeled with Fitts' law relationships, for example in (Crossman & Goodeve 1983), but not in movements involving combinations of both. Though simplistic, the choice of only 2 DOF was judged a necessary first step towards modeling more complex 6 DOF movements, and also allows for direct comparison with the same translational and rotational movements performed separately.

But how can a model for combined movements be deduced from the Fitts' law relationships of the separate movements? One potential approach is shown in Equation (2.6), where the mean time for a combined movement is assumed to be equal to the sum of the times of the separate movements. This means that the completion time for a task where both rotation and translation are performed as a combined movement and the total completion time when the movements are performed separately should be approximately equal.

$$MT_{combined} = MT_{rotation} + MT_{translation} \tag{2.6}$$

However, a further simplification of the model would be beneficial to reduce the number of experimentally determined values. As mentioned earlier, previous studies have indicated that rotational movements have a similar index of performance as translational movements (Crossman & Goodeve 1983). If true, the combined model could potentially be reduced to that seen in Equation (2.7). The rotational distance equivalent here is α, while ω represents the rotational tolerance. This is a simple linear model with two experimentally determined coefficients like Fitts' original law, and with an index of difficulty that is fully defined by the four parameters describing the combined task. From this a combined mean of means TP can also be calculated, which describes the speed and accuracy of the combined movement in one dependent variable.

From an application point of view this would make it an attractive model for movements including both rotations and translations.

$$
\begin{aligned}
MT_{combined} &= a_{combined} + b \cdot [ID_{rotation} + ID_{translation}], \\
ID_{rotation} &= log_2\left(\frac{\alpha}{\omega} + 1\right), \\
ID_{translation} &= log_2\left(\frac{A}{W} + 1\right).
\end{aligned}
\tag{2.7}
$$

Three experiments were performed to evaluate the model for combined movements in Equation (2.7). Experiments 1 and 2 were straightforward applications of Fitts' law for separate 1 DOF translation and 1 DOF rotation, respectively. These then formed the basis for comparison with the combined movements with 1 DOF translation and 1 DOF rotation performed in experiment 3. The three experiments each used a separate set of participants. These were 12, 13 and 13 students and staff of the University of Maryland respectively. The three experiments are outlined below, see Appendix A for the full details on participants, apparatus, stimuli and procedure.

2.3.4.1 Experiment 1: Translational Movements

This experiment involved 1 DOF translational movements only, the standard application of Fitts' law. There were 16 levels of ID, via combinations of 4 translational distances and 4 translational tolerances. See Table A.1 in Appendix A for details.

Method The experiment consisted of 12 participants performing translational movements of a virtual object. See Fig. 2.6a for the task, and Appendix A for details.

Results and Discussion The mean completion time across participants for different IDs can be seen in Figure 2.7. The relative standard deviation ranged from 20.7% for an ID of 3.09 bits, to 30.6% for an ID of 1.52. A linear fit produced a slope of 0.31 and an intercept of 0.22. The square of the correlation coefficient for the fit, r^2, was 0.984. 2.3% of trials were rerun due to participant error. The mean of means TP was 3.17 bits/second. The correlation to the linear fit postulated by Fitts' law was similar to that typically reported in the literature for 1 DOF translational tasks ((Fitts 1954); (MacKenzie & Buxton 1992)), in spite of the small but noticeable time delay inherent in the mapping from the sensor to the cursor on the display. Equivalent Flock of Birds systems have been found to exhibit tracker latencies from movement to system response of 23 ms (Mine 1993).

2.3.4.2 Experiment 2: Rotational Movements

This experiment involved rotational movements only. 16 levels of ID were used, through combinations of four rotational distances and four rotational tolerances. See Table A.1 in Appendix A for details.

Method The experiment consisted of 13 participants performing rotational movements of a virtual object. See Fig. 2.6b for the task, and Appendix A for details.

Results and Discussion The mean completion time across participants for different IDs can be seen in Figure 2.8. The relative standard deviation ranged from 21.9% for an ID of 3.46 bits, to 32.2% for an ID of 4.79. A linear fit produced a slope of 0.32 and an intercept of 0.27. The

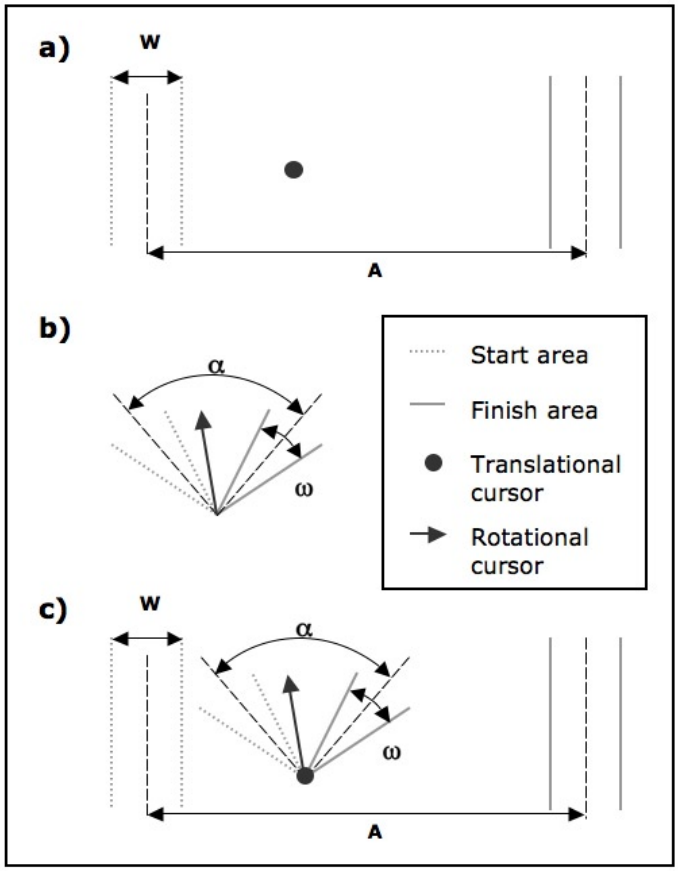


Fig. 2.6: *Representations of visual stimuli provided to the participants for Experiments 1, 2, and 3 (a, b, and c, respectively, in figure). Task variables have been added for clarity but were not part of the display shown to the participants in the experiments.*

square of the correlation coefficient for the fit, r^2, was 0.930. 1.7% of trials were rerun due to participant error. The mean of means TP was 3.02 bits/second.

A reasonable correlation to a linear fit was achieved, although less so than for translational movements (for a 95% confidence interval). This can also be seen from Figure 2.8, where the results from changes in distance do not line up as well with those from changes in tolerance. However, the slope of the linear fit was very similar to that found for translational movements (differing by 4.4%). Similarly, the mean of means throughput is 4.9% lower than in experiment 1. This, together with evidence from the literature (Crossman & Goodeve 1983) indicates that these types of translational and rotational movements indeed have approximately the same index of performance, supporting the model proposed. There is a 24.3% increase in the intercept of the fit for the rotational movements; however, this measure is typically attributed to non-informational aspects of the task (Zhai 2004). Thus it is not affected by the task distance or tolerance, but rather may indicate the cognitive effort required for initiating a movement, or regression errors.

2.3.4.3 Experiment 3: Combined Movements

This experiment involved movements with a translational and rotational component. 16 levels of the combined ID were used, with combinations of two rotational distances, two rotational tolerances, two translational distances and two translational tolerances. The model presented in Equation 5 was used to establish the indices of difficulty. The sum of the rotational and

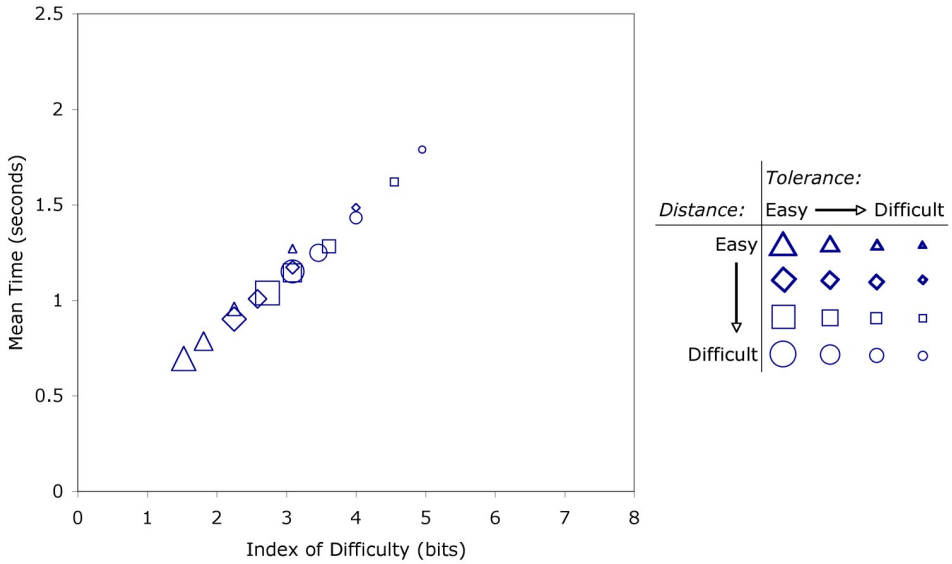


Fig. 2.7: *Mean completion time against index of difficulty for translational task in Experiment 1. The levels of tolerance (here, W) and distance (here, A) are indicated for each task.*

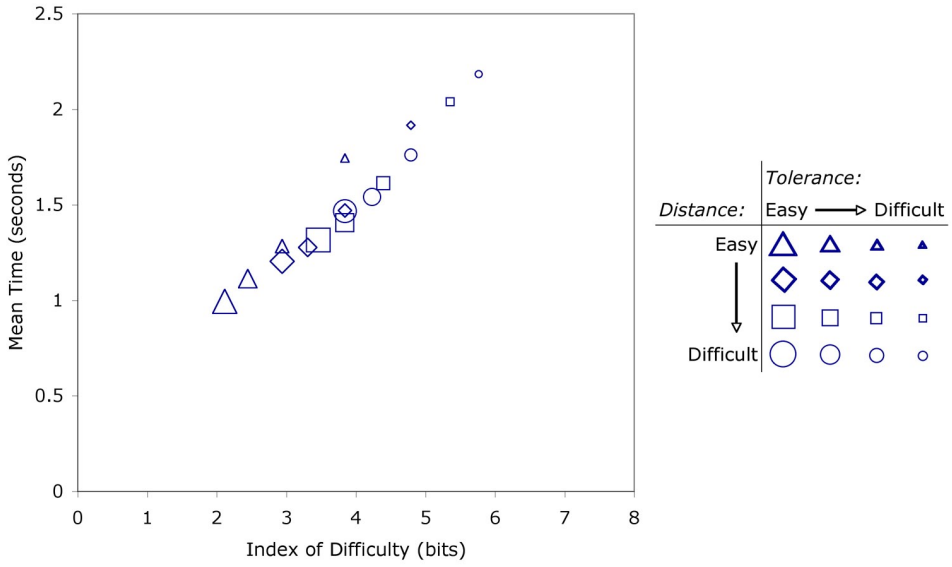
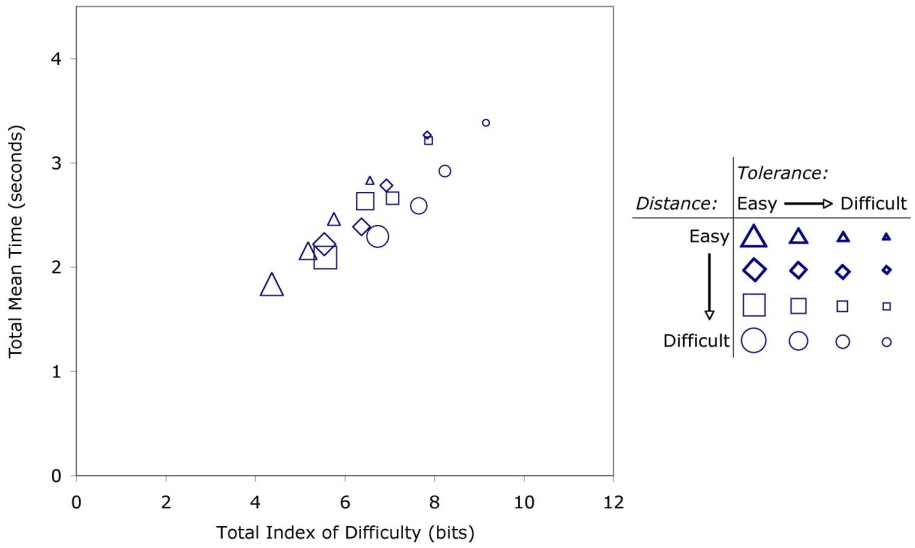


Fig. 2.8: *Mean completion time against index of difficulty for rotational task in Experiment 2. The levels of tolerance (here, ω) and distance (here, α) are indicated for each task.*

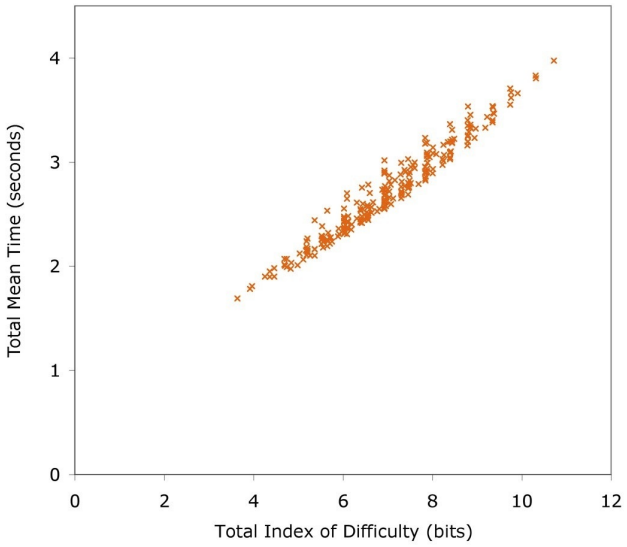
translational IDs ranged from 4.37 to 9.15 bits. See Table A.1 in Appendix A for details.

Method The experiment consisted of 13 participants performing combined rotational and translational movements of a virtual object. See Fig. 2.6c for the task, and Appendix A for details.

Results and Discussion The mean completion time across participants for different indices of difficulty can be seen in Figure 2.9a. The relative standard deviation ranged from 21.8% for an ID of 5.54 bits, to 35.1% for an ID of 9.15. A linear fit produced a slope of 0.32 and an intercept of 0.46. The square of the correlation coefficient for the fit, r^2, was 0.817. 2.8% of trials were rerun due to participant error. The mean of means TP was 3.28 bits/second.



(a) *Combined movements.*



(b) *Separate movements.*

Fig. 2.9: *Mean completion time against total index of difficulty for combined task in Experiment 3 (a in figure) compared with summed separate rotational and translational results from Experiments 1 and 2 (b in figure). The levels of tolerance (here, W and ω) and distance (here, A and α) are indicated for each combined task.*

A lower correlation was achieved as compared to experiment 1 but not as compared with experiment 2 (for a 95% confidence interval). Thus the index of performance for combined movements has a stronger dependency on the composition of the distances and tolerances of the task than for simple translational movements. Although there is still a clear linear trend,

this increased dependence means the data only provides partial support for the model proposed. However, the combined slope matched the slopes found in the two first experiments very well, differing by 3.9% from the slope of the translational movements in experiment 1 and 0.5% from the slope of the rotational movements in experiment 2. The mean of means throughput is 3.2% higher with respect to experiment 1 and 8.6% higher with respect to experiment 2.

In Figure 2.9b the results for the separate movements in experiment 1 and 2 were included for comparison. This was done by adding the mean times and respective indices of difficulty for each combination of separate rotational and translational movements to produce 256 points. Thus the separate movements in the figure represent the application of the combined model (Equation (2.7)) with data from experiment 1 and 2. A linear fit to this data ($r^2 = 0.953$) produced a slope of 0.32 and intercept of 0.5. This was very close to the result obtained for the combined movements, differing by only 1.5% in slope and 7.8% in intercept. These results indicate that the index of performance for combined movements on average may be equivalent to that for separate rotational and translational movements.

A four-way analysis of variance (ANOVA) of the results in experiment 3 was performed to assess the contribution of the four task parameters on the results obtained. A mixed model was used, with the two distances (low, high) and two tolerances (low, high) as fixed effects, and participants as a random effect. Thus it was assumed that the participants were sampled at random from a large population and could be modeled as a random variable with zero mean and an unknown variance.

All the main effects were significant on the .05 significance level, yielding F values of $F(1, 2468) = 51.6$, $p < .0001$, eta squared = .010, for rotational distances, $F(1, 2468) = 170.2$, $p < .0001$, eta squared = .033, for translational distances, $F(1, 2468) = 804.0$, $p < .0001$, eta squared = .154, for rotational tolerances and $F(1, 2468) = 288.9$, $p < .0001$, eta squared = .055, for translational tolerances. Thus both distances and both tolerances significantly affected the time taken to perform the combined task, supporting their inclusion in the model. In addition there were two significant interactions on the .05 significance level, though with relatively low effect size indices.

The interaction between the two distance parameters yielded $F(1, 2468) = 17.1$, $p < .0001$, eta squared = .003, while the interaction between the two tolerance parameters of the task yielded $F(1, 2468) = 8.5$, $p < .005$, eta squared = .002. In other words, the execution of the rotational movement was affected by the distance and tolerance requirement on the translational movement, and vice versa. This indicates that the participants performed some level of coordinated planning and execution of the two movement components. Indeed, it was found that twelve out of thirteen participants performed the combined movements in parallel, while one performed them strictly serially. Examples of combined trajectories for a single task (one out of twelve occurrences) across all thirteen participants can be seen in Figure 2.10. The same pattern was observed across all trials.

To explore this coordination further, a numerical comparison of the relative timing of the rotational and translational movements across all thirteen participants was performed. A new measure was introduced: the difference in time at which the participant first crossed the halfway point between the translational and rotational start and finish areas, Δt_{mid}. The halfway point was used to avoid the ambiguity in coding the often multiple crossings of the finish area. The two sampled points flanking the actual boundary were used, and a best estimate of the actual time was found using linear interpolation. The mean of the absolute value of Δt_{mid} across all trials for the 12 participants that performed the movements in parallel was 114 ms (participant means ranging from 67 ms to 247 ms), with a standard deviation of 121 ms. The mean of the absolute value of Δt_{mid} for the participant that performed the movements serially was 1620 ms with a standard deviation of 816 ms.

Thus an order of magnitude difference in the relative timing of translational and rotational

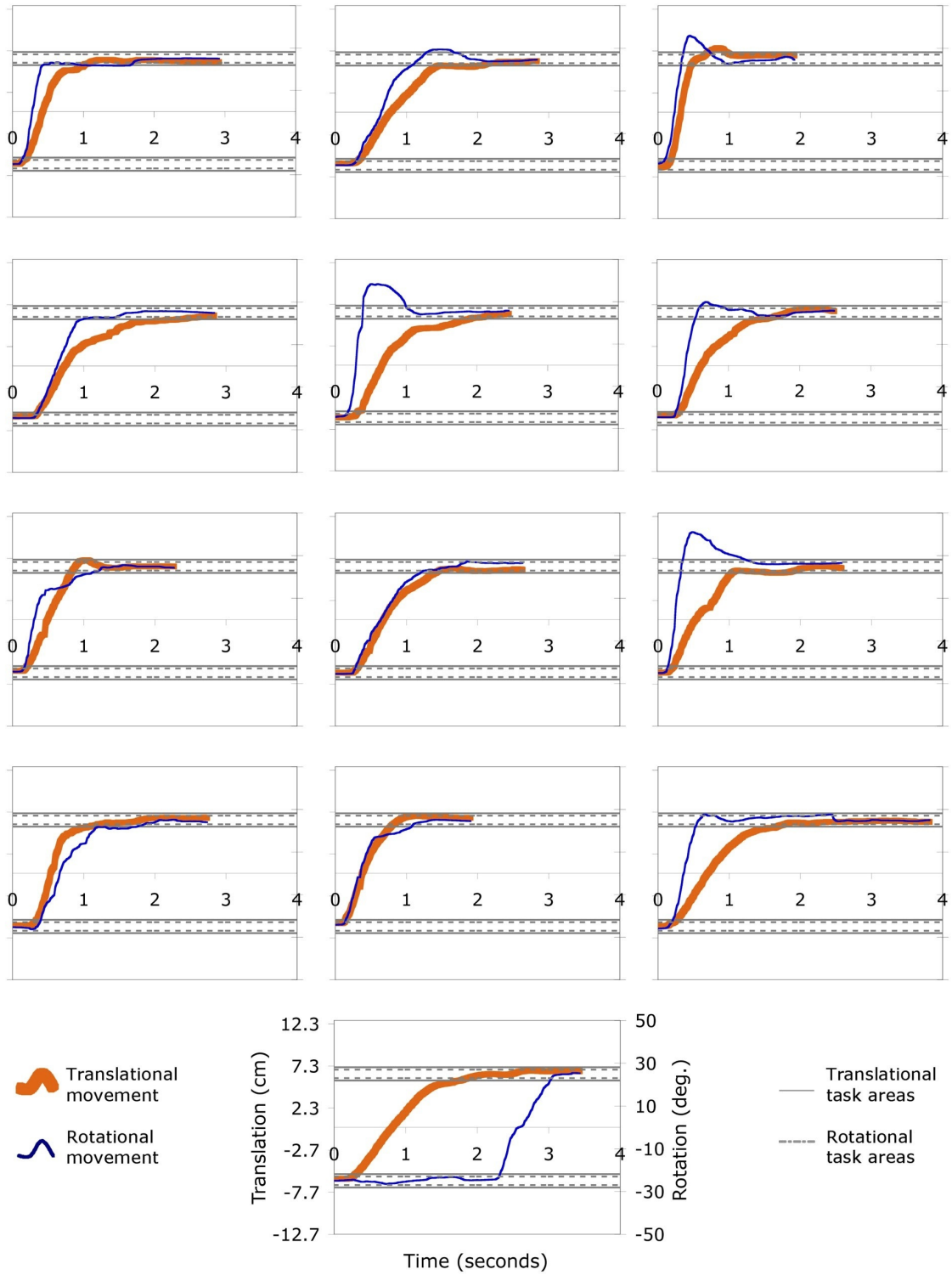


Fig. 2.10: *Sample trajectories for 13 participants on a combined task ($A = 12.7$ cm, $W = 1.6$ cm, $\alpha = 50°$, and $\omega = 4°$). Axis scales, labels and units are the same for all plots.*

movements was observed between the participants performing in parallel and the one in serial. Few conclusions can be drawn about the latter without more data however. This participant was one of nine that had previous experience with input devices for 3D applications, so this is not a likely reason for the difference in execution. For the twelve participants performing the movements in parallel, the rotations and translations were executed with a surprisingly high degree of coordination across participants, despite the physically disparate nature of the two movements. In addition, the two movements had for many trials very different indices of difficulty and expected completion time, but seemed to be executed so as to start and finish at the same time. Thus for these participants the execution can be said to be strictly in-phase. The literature on pattern generation in the central nervous system has rich observations of phase interactions in a diverse set of movements. A classic example is the switch from anti-phase to in-phase coordination at a given frequency in simultaneous bilateral finger oscillations (Kelso 1984). Although these oscillatory movements are different from the discrete tasks performed in this study, the idea of coordinating diverse movements using a central rhythmic unit may be applicable.

2.3.4.4 General Discussion

One of the main concerns when designing the three experiments was the combination of distances and tolerances used. It was desirable to be able to compare the results from experiment 3, using the combined model presented in Equation 5, directly with the results from the first two experiments. Another constraint was the size of the virtual object for the combined movements in Figure 2.6c, which limited the translational movements to about 20 cm. The resulting ID levels were therefore relatively low, with several values below 3, as seen in Table A.1. (Gan & Hoffmann 1988) found that a linear model relating MT to the square root of the distance provided a better fit than Fitts' original law (Equation (2.3)) for ID values below 3. However, the same distance and tolerance combination gives an equivalent ID of only 2.322 with the Shannon formulation used in this study (Equation (2.4)). Thus 75% of the ID values for the translations in experiment 1 and 3 were above the threshold, while the corresponding percentages for the rotations in experiment 2 and 3 were 93.75% and 100% respectively. This probably indicates a sufficient manipulation of IDs for the purpose of this study.

Another design decision made was to only reflect the DOF used for each experiment on the display. In addition, the translational and rotational DOF that did not have a requirement for distance and tolerance were not physically constrained. Thus the participants were, for example, allowed to perform rotations while moving the hand-held sensor in experiment 1, but only left to right (and vice versa) translations of the cursor were displayed on the display. Constraining the movements physically would be feasible for the simple 2 DOF movements performed in this study, but was avoided to enable direct comparison with potential future extensions to movements with several translational and rotational DOF. Constraining these more complex movements to exactly the DOF of interest may prove difficult. In addition, such constraints typically do not exist in the high-DOF input devices for which the work presented here is intended. Another option considered was to provide a graphical representation of all translational and rotational movements on the display, to exactly represent the pose of the hand-held sensor. A graphical representation used frequently for 6 DOF docking tasks is that of one tetrahedron that is to be aligned with an equal size target tetrahedron (see for example (Zhai & Milgram 1998)). However, it is not clear how to represent clearly the tolerance required on the specific rotational or translational DOF used in this type of 3D representation. Other issues include occlusions and the need to provide some form of depth perception. Although not directly relevant to this study due to the constrained number of degrees of freedom, these issues will need to be dealt with for future extensions of the model to high-DOF movements.

The slope coefficients found for experiment 1, 2 and 3 were 0.31, 0.32 and 0.32 respectively with the Shannon formulation (Equation (2.4)). This corresponds to an Index of Performance (IP) of 3.2, 3.1 and 3.1 bits/second respectively. The mean of means throughput (TP) performance measures were similar, namely 3.17, 3.02 and 3.28 bits/second. In comparison, Fitts' original study (Fitts 1954) with reciprocal hand movements reported an IP of 10.6 bits/second. Another famous example is that of (Card et al. 1978), which reported an IP of 10.4 bits/second for a mouse in a text-selection task and 4.9 bits/second for a joystick. In general, the IPs reported vary drastically between studies, although most are in the range of 3 bits/second to 12 bits/second (MacKenzie 1992). Indeed a more recent survey included nine ISO conforming studies that reported a mean of mean TP of 0.99-2.9 bits/second for touchpad devices and 3.7-4.9 bits/second for mice (Soukoreff & MacKenzie 2004).

In summary, it was found that the time taken to complete a movement consisting of 1 DOF rotational and translational components was equivalent to the sum of the time taken to complete each component separately. Thus two Fitts' law relationships, one for the rotational part and one for the translational part, can be used to represent the combined movements. However, it was also confirmed that rotational and translational movement have a similar IP and mean of means TP, enabling the proposed simplified model for combined movements in Equation (2.7). This model was found to provide a reasonably accurate estimate of the mean time for a combined movement and to allow for the estimation of a single combined IP (and mean of means TP) for combined movements. The model proposed thus provides additional value in that it can be used to compare input devices over a range of rotational and translational requirements with a single dependent variable. The model is not a new version of Fitts' law, but rather a proposal for how to combine the Fitts' law estimates made for the rotational and translational movement components. Further improvements in the accuracy of modeling these movements can probably be made by increasing the number of task parameters, or by introducing more empirically determined constants that take into account the observed interactions between the translational and rotational movements. However, this may also reduce the scope of tasks for which the model can be used and introduce additional requirements for experimental data. It is hoped that the model in its current form can be useful to human factors practitioners that deal with combined rotational and translational movements, while being as simple to apply as Fitts' original law. For assistive manipulators the immediate benefit is the knowledge that Fitts' law also governs, at least partly, the combined movements required for controlling such robots.

2.3.5 Metric: Mean Time with Collisions Included

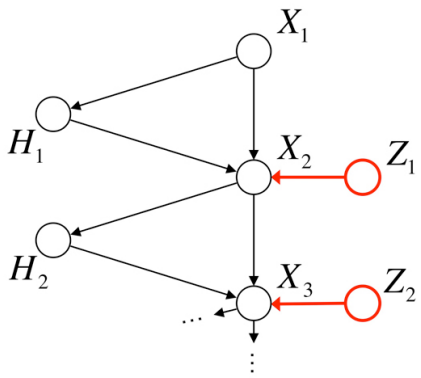
When using Fitts' law the participants are instructed to maintain an error rate below a certain level (around 2-5%). This is critical to disambiguate results with low times but a high number of errors, and vice versa, but is difficult for a participant to control on more complex tasks. For the high-DOF movements involved in teleoperating assistive manipulators there are few models available, however the necessity of trading off velocity with accuracy remains. In this type of experimental setting the trade-off can be quantified by the MT to complete a task and the number of collisions between the manipulator and the environment, respectively. For the experimental evaluations performed in Chapter 4 of this thesis, the consequences of an error has therefore been included in the time to complete the task, see experiment protocol sections. That is, each collision with the environment costs time and the participants are given a motivation to minimize the MT. Put in the context of the signal-dependent noise in the human sensorimotor system (Faisal et al. 2008), the user has to adjust the speed of execution to achieve a variance of the trajectories performed (variance of X in Fig. 2.2) that statistically minimizes the average MT over attempts. Given that a higher velocity will lead to a higher variance, which also means a higher probability of colliding and loosing time.

2.3.6 Discussion

Several theories and models exist for explaining the speed-accuracy trade-off inherent in human movements, and that is clearly visible in Fitts' law tasks. For the practitioner, however, most of these theories are not easy to apply in the two main uses of Fitts' law, comparing input devices and making movement time predictions. This ease of application to experimental research ensures that Fitts' law will likely remain a useful tool in the design of human-machine interfaces for the foreseeable future. The above extension of Fitts' law to combined movements can therefore have implications in evaluating human-machine interfaces for assistive devices intended for users with disabilities and special needs. In fact, the original Fitts' law is currently being applied to quantify performance of new Brain-Computer Interfaces (BCI), for example (Felton et al. 2009), that are intended for users with severe disabilities. These interfaces are currently very limited in throughput, and are thus usually only applied to low-DOF tasks (Tonet et al. 2008). Among the other potential applications could be the expansive field of gesture recognition for mobile devices, driven by the inclusion of accelerometers and other movement sensors into mobile phones and portable music players. Here physical movements of the device itself are interpreted and converted to interface actions, for example using 2 DOF rotations to navigate a graphical user interface, see (Crossan & Murray-Smith 2004). Combining rotations and translations could potentially increase the vocabulary of gestures significantly. However, establishing suitable requirements for speed and accuracy of these movements would be made simpler by the use of a model like the one presented here.

For researchers the finding of a high degree of coordination between the translational and rotational component could be of interest. One potential approach could be to investigate the effect of the instructions given to the participants. For example, future experiments might explicitly specify that the components should be executed in series or in parallel. Another approach could be to compare the results obtained here with an equivalent bimanual experiment. Would the same degree of coordination be observed if the rotational and translational component were split between the two hands? Could this lead to insights into whether to separate the rotational and translational components in high-DOF input devices? For example for assistive manipulators? The model was tested with a free-flying input device, however Fitts' law has been shown to be robust across a diverse range of input devices in the past. Does this also extend to combined rotational and translational movements? I also believe that further extensions to the model can be developed for more complex movements. Empirical work is important for determining the most effective use of high-DOF user interfaces as they continue to evolve (Bowman et al. 2004). With a model for high-DOF movements the comparison of these interfaces will be less ambiguous and should allow for generalization beyond a specific task. It is hoped that extensions of the work performed here may one day provide a theoretical basis for modeling complex high-DOF tasks like virtual reality navigation and robot teleoperation.

The confidence that Fitts' law can be applied to more complex movements also led to the definition of a new metric that is directly applicable to assistive manipulator tasks. While not directly following the Fitts' law paradigm with established distance and tolerance parameters, the inclusion of the time cost of collisions in the Mean Time measured greatly reduces the ambiguity in the results. See Chapter 4 for an application of this metric.



2.4 Modeling and Simulating Physical Disabilities

2.4.1 Introduction

Real end-user participants are needed to validate the clinical credibility of any assistive technology (Tsui & Yanco 2009). Moreover the technical development of the assistive tools needs to be driven and validated from the beginning by test cases that are as realistic as possible. From the ethical standpoint, the involvement of persons with disability in a lengthy and tiring systematic activity of validation in intermediate stages of development must be reasonably limited. From the practical standpoint, the set of disabilities and abilities of a given user group for an assistive manipulator can be large and diverse. It can for example be hard to find a homogeneous set of disabled participants. This has implications for the experimental evaluations of such systems and their development. Experimental comparisons can be made more reliable and easier to replicate by simulating a consistent disability for a set of able-bodied participants, by introducing controlled perturbations in the perception-action loop. This shortens the prototype development and facilitates the experimentation on more specifically robotics research issues. However, the simulation of disabilities in able-bodied persons is not obvious, as the neurophysiological modeling of disabilities is still a research issue.

2.4.2 Characterizations and Measures of Disabilities

Pathological tremor is the most typical movement disorders, in one study reported to affect 14.5% of persons between 50 and 89 years (Wenning et al. 2005). The same study also reported other movement disorders, such as restless legs syndrome (at 10.8%), parkinsonism (at 7%). The most common form of tremor is Essential Tremor (ET), which is an involuntary and rhythmic oscillatory movement with a frequency of 4-12 Hz that primarily affects the arms of the patient (Rubchinsky et al. 2007). ET is thought to affect around 5% of persons over 65 years of age (Louis & Ferreira 2010), and between 5 and 10 million persons in the United States (Pahwa & Lyons 2003). Patients with tremor may benefit from an assistive manipulator, for example if the patient is unable to perform ADL, but a physical or medicinal inhibition of the tremor may often be more beneficial. See for example (Gallego et al. 2013).

The most suitable users for assistive manipulators seems to be those with severe disabilities in the upper limbs that cannot be treated or inhibited. Here there are less clear classifications of syndromes, but the list will in general include persons that have either had damage to the spinal chord from an accident, or that is suffering from a muscular or neural disorders. Partial or total paralysis of the limbs and torso is also known as tetraplegia. It has been reported that approximately 6 million persons are living with paralysis in the United states, of which 23% result from spinal cord injury (Kim 2012). Often users of powered wheelchairs are persons

that can also benefit from an assistive manipulator. A 1990 survey on powered wheelchair users found that the most prevalent disabilities were spinal cord injury (at 24%) and multiple sclerosis (16%) (Prior 1990). Multiple Sclerosis (MS) is a disease that interrupts the communication in the nervous system by damaging the insulating cover of nervous cells. Patients that have had a stroke are also candidate users. On the extreme end of the disability scale we find patients with the locked-in syndrome. Such patients have complete paralysis of all voluntary movements, except for the eyes. Although controlling an assistive manipulator is therefore also difficult, great progress has been made on invasive Brain-Machine Interfaces (BMI). For example allowing a patient to grasp objects with a robot arm using direct signals from electrodes implanted in her brain (Hochberg et al. 2012).

There also exists several measures for characterizing the functionality of patients with disabilities. Such measures can be used to get a better idea of the ability of a patient to perform the tasks needed for his/her daily life. Functional performance measures are important parts of end-user evaluations of assistive technologies in general, and the Functional Independence Measurement (FIM) tool has been suggested as particularly suitable for assistive robot arms (Tsui et al. 2009). The FIM includes scores from 1 (total assistance) to 7 (complete independence) in among other eating, grooming, dressing, and transferring to/from a wheelchair. See (Dodds et al. 1992) for a validation of the tool.

2.4.3 Disabilities in Experimental Evaluations

Several experimental evaluations of assistive robots exist in the literature, see (Tsui et al. 2009) for an overview. For example the testing of the EL-E assistive robot in picking up objects from the floor by 8 participants with Amyotrophic Lateral Sclerosis (ALS) (Choi et al. 2008). Another example is an experimental evaluation of the Raptor arm with 11 participants with spinal chord injury on 16 ADLs (Chaves et al. 2003). The tasks included among other picking up and drinking from a cup, opening and accessing objects in a refrigerator, operating a toaster, and picking up keys from the floor. However, as mentioned above, experimental evaluations with disabled end-users can be time consuming and it may be difficult to achieve a homogeneous set of participants. Using able-bodied users can be an alternative in evaluations of novel technologies, at least for guiding the development up until the stage where full end-user testing is feasible. Simulating a disability is a good way to increase the realism of able-bodied user trials and to increase the validity of the results also for disabled users. Some specific disabilities like tremor can be modeled directly, but is also less relevant here as it can be removed or compensated through adaptive filtering techniques (Rocon et al. 2012). When a specific model cannot be obtained, or when it is not appropriate to focus on only one disability, a complementary approach is to assume a generic physical disability. Previous examples of this approach include assessments of computer mouse movements by disabled persons (Mankoff et al. 2005) and of shared control with joystick input for assistive wheelchairs (Vanhooydonck et al. 2010). In (Mankoff et al. 2005) a random noise was added to the user input, while in (Vanhooydonck et al. 2010) the user was prevented from commanding movements to the left/right (by a deterministic effect) or from specifying the exact movement direction.

2.4.4 Simulated Disability: Signal-Dependent Gaussian Noise

For the experiments performed in Chapter 4 a Gaussian noise was added to the raw user input (Z in Fig. 2.3), in analogy with (Mankoff et al. 2005). The noise was filtered to below 2 Hz to be comparable to typical human movements in the frequency domain. In order to have a more realistic simulation it was also made to increase in strength with the magnitude of the velocities commanded by the user. This emulates the effect already seen in the speed-accuracy

trade-off of many human movements, as mentioned above. That is, faster movements require greater forces in the muscles, which again introduces more neuromotor noise (Faisal et al. 2008). An increase in this signal-dependent neuromotor noise has been related to stroke-related motor deficiencies (McCrea & Eng 2005), and children with dystonia (movement disorder that causes involuntary muscle contractions) (Sanger et al. 2005). The quality of the speed-accuracy trade-off is also reduced in children with cerebral palsy (reduction in ability of brain to transmit nerve signal to the muscles) (Smits-Engelsman et al. 2007). Note that a disability can also require the use of less dexterous parts of the body for robot control, for example movements of the feet or head. Such movements are also subject to the speed-accuracy trade-off, see for example (Drury 1975). While the noise added does not necessarily correspond exactly to a given real disability, it is an attempt at emulating the negative effect a disability could have on the ability of the user to accurately control the manipulator and of the robot control system to effectively react to them. In the context of the information-theoretic model presented in Section 2.2.3, we can see a disability as a generic disturbance which has the effect to reduce the mutual information over the human input H and the robot actuation A.

2.4.5 Discussion

Given the large variety of disabilities that exist, assistive manipulators have to cater for a wide audience. That is, there is likely no one-fits-all solution to disabled persons achieving independence on ADL through a robotic device. There are approaches for directly inhibiting specific disabilities like tremor, helped by the fact that such a disability induces movements that are of higher frequencies than most human intentional movements. The work presented here will focus on the cases where an explicit model of the disability is difficult to obtain, and where the task is rather to try to overcome a reduction in the ability of a user to control the robot. The evaluations are here primarily with able-bodied users, where a generic signal-dependent Gaussian noise is used to simulate just such a situation. The performance metrics used are therefore also mainly quantitative. In testing with real end-users the gap between such measures and the functional performance measures should be attempted filled (Tsui et al. 2009). Evaluating the physical and cognitive workload is also a good idea, for example using the NASA-TLX subjective workload measure. Especially as an assistive manipulator is a tool aimed at daily usage, and where each task can take much longer than for an able-bodied person. The NASA-TLX is for example used in Chapter 5 of this thesis.

2.5 Scientific Method, Benchmarking and Replicable Experiments

There has recently been an increased emphasis on improving the experimental method in robotics. This includes yearly workshops on benchmarking at the major robotics conferences, guidelines (Bonsignorio et al. 2009), reviews of experimental practices (Amigoni et al. 2009), and open data repositories (Bonarini et al. 2006), (Howard & Roy 2003). Consistent methodologies for benchmarking are essential for comparing different solutions, and for driving the robot development process, and can help speed the process of getting assistive technologies to the level of clinical trials and out to the end-user (see (Tsui & Yanco 2009) for related issues). It may also promote replicable experiments for this type of human-robot systems.

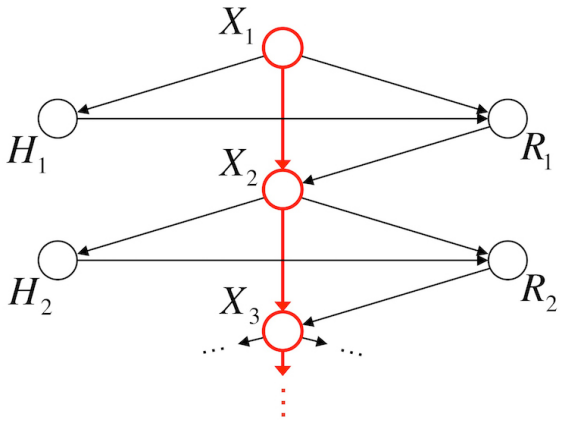
Given that much of the results gathered for this thesis are from coupled human-robot systems, an equivalent user group operating a similar robot embodiment in a similar experimental setting is required to compare performance with other approaches. That is, the adaptation of both agents will affect the trajectory of development, and neither one can be assumed fixed. A first open repository for sharing similar experiments was therefore created at http://throughput.sourceforge.net. Several of the experimental evaluations proposed here are

posted in this repository, to enable other researchers to verify, compare and extend the results. Each includes: a) all data gathered; b) the complete set of executables for running each experiment; c) instructions for installation and experiment protocol; d) templates for loading and analyzing the data gathered (MATLAB/Octave).

2.6 Discussion and Conclusion

The work described in this Chapter has been essential for defining the general approach of the thesis, which is somewhat untraditional from a robotics perspective. First of all, the emphasis on studying the complete system as a coupled human + robot binomial with information-theoretic concepts. Other approaches to modeling may also be fruitful, but the one presented here provides insights and quantitative metrics that has helped drive the thesis forward. It is especially interesting to see the close relationship to the Information Theory roots of Fitts' law and the hypothesized signal-dependent noise in the neural system. An important part of the Chapter has been a study that to the author's best knowledge is the first in showing that Fitts' law can be used to model combined rotational and translational movements. The related speed-accuracy trade-off of human movements has also driven the development of a shared control system for assistive manipulators (see Chapters 4, 5 and 7), and has been used to define a simulated disability that has some grounding in the sensorimotor system and in certain disabilities. This has been important for running extensive controlled experiments with able-bodied users. It is believed that the effort of making the experimental evaluations of the thesis replicable also represents a novelty and a contribution to the field.

3. ADAPTING THE PHYSICAL ROBOT DESIGN TO THE USER'S TASKS



3.1 Introduction

Designing the physical structure of an assistive manipulator can be a complex undertaking, as such robots are typically expected to perform a diverse set of tasks in complex environments, like a user's home. The current version of ASIBOT has been used for several years and is under constant development. A process for evaluating a new design of this robot was the drive for the developments in this Chapter. The work presented is focused on optimizing the kinematics of such a symmetric assistive climbing robot for enabling a broad range of assistive tasks. The kinematics is important as it affects not only the tasks achievable, but also the structural requirements of the robot and the support points in the environment for climbing. It also affects the user acceptance of the final system. For example, a smaller robot can more easily be transported.

3.2 Task-Oriented Design of Robots

Park, Chang and Yang (Park et al. 2003) attempted a Task-Oriented Design (TOD) approach for the kinematics of an assistive manipulator, that would guarantee that at least a set of tasks with high priority would be possible in a given environment. The end-result should then also be capable of performing tasks that are similar. The approach begins with an investigation into potential tasks, including robot base location, environment obstacles and Task Points (TP). The latter is used to denote the position and orientation requirement of the robot end-effector to achieve a given task. This is followed by an optimization of the kinematic design using the information from the task analysis. For this a grid-based method was developed and applied to an assistive non-symmetric robot mounted on a mobile base (Chang & Park 2003). The basic concept of this Grid Method for kinematic optimization is to represent each joint as a unit grid,

each of which can be treated as much simpler individual optimization problems using mainly information from neighboring unit grids. One of the objectives of this chapter was to extend this method for kinematic optimization to symmetric robots like ASIBOT.

Several other methods for optimizing kinematics exist, however the Grid Method was chosen at it has been shown to be very efficient for solving problems with high-DOF robots and many task points. Assistive robotics typically involve robots from 5 to 7 DOF designed for a large, diverse set of tasks. (Kim & Khosla 1993) proposed a comparatively complete algorithm for optimizing the design of general manipulators based on a genetic algorithm and applied it to the design of a servicing robot for the Space Shuttle. This was based on a task-based design approach similar in concept to TOD. (Chocron & Bidaud 1997) developed a method based on combining modular segments of a fixed number of types into a modular robotic system.

However, most approaches that do not use grids require the calculation of the inverse kinematics of the robot during optimization, which adds to the computational load. Another issue is the number of design variables required. Kinematic optimization typically involves a given number of design variables per joint, for example the 4 Denavit-Hartenberg (DH) parameters (see section 3.3.1 for the definition of DH parameters used here). For a robot with n joints optimized for m task points this means $4 \cdot (n\text{-}1) \cdot m$ design variables are needed to optimize the complete robot for all desired tasks. This is a problem as the search space increases exponentially with the number of design variables (Park et al. 2003). The Grid Method on the other hand has a fixed number of design variables, 4 for each unit grid, regardless of the number of DOF and task points used. The drawback is that the optimization has to be performed for each unit grid, however the number of unit grids varies linearly with both DOF and task points. In general it also does not require the calculation of the inverse kinematics of the robot during optimization. This implies that the method can more easily be scaled to complex problems.

3.3 Method

The Grid Method and the modifications made to be able to optimize symmetric robots are described in section 3.3.2, after the description of the kinematic structure of ASIBOT in 3.3.1. A methodology for designing the kinematics of a climbing assistive robot is then introduced in section 3.3.4. The methodology was applied to a real kitchen environment, and a quantitative virtual evaluation of the performance of the robot designs found was made.

3.3.1 ASIBOT Kinematic Structure

When describing the kinematic structure of a robot several notations can be followed. The DH notation of Paul (Paul 1981) is here used, which can be seen in Fig. 3.1. For a kinematic optimization approach this notation allows for writing constraint functions in terms of four parameters that are relatively easy to visualize, joint angle θ, twist angle α, link length l and link offset d. The same notation was also used in the earlier work on grid-based kinematic optimization.

The current design of ASIBOT can be seen in Fig. 3.2, with the DH parameter interpretation shown in Table 3.1. Note that parameteres change depending on the docked side. The $DS{+}MU$ parameter is the sum of DS lenght and the first link docked, BR is the lenght of each central link and $MU + EX_1$ and $MU + EX$ are the lengths of the last links with and without the gripper's fingers. The actual values for DS, MU, BR, EX and EX_1 are 162, 148, 400, 120 and 180 mm, respectively. The maximum reach without extendending the gripper's fingers is 1378 mm, and is 1438 mm in extended finger state.

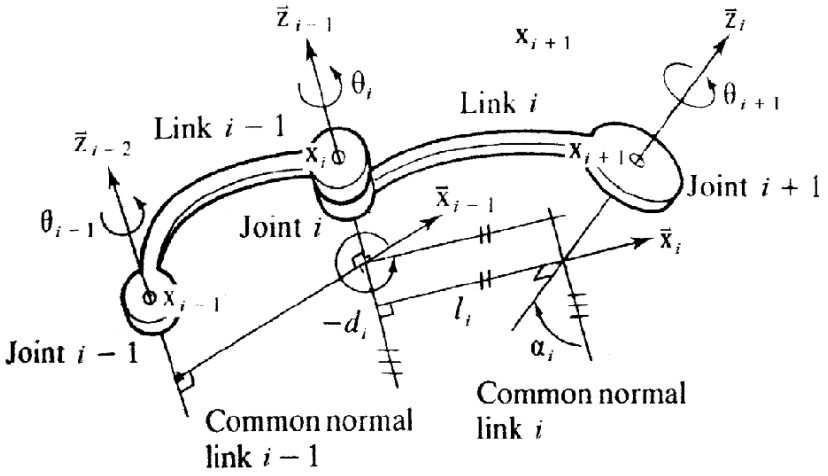


Fig. 3.1: *Definition of Denavit-Hartenberg (DH) parameters used (Paul 1981).*

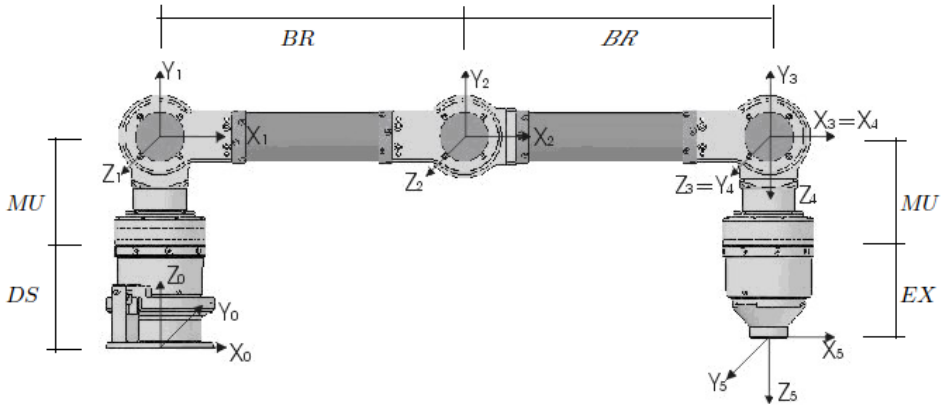


Fig. 3.2: *Current ASIBOT robot design with lengths and frame assignment.*

Tab. 3.1: *Denavit-Hartenberg (DH) parameters of the current ASIBOT robot.*

Joint	θ	l	d	α	*Range*
1	$\theta 1$	0	*MU+DS*	90	360°
2	$\theta 2$	*BR*	0	0	270°
3	$\theta 3$	*BR*	0	0	270°
4	$\theta 4$	0	0	90	270°
5	$\theta 5$	0	*MU+EX*	0	360°

3.3.2 Modified Grid Method for Symmetric Robots

3.3.2.1 Original Approach

The algorithm described here is based on the Grid Method for optimizing robot kinematics using the Very Fast Simulated Annealing (VFSA) optimization method, as presented in (Park et al. 2006). See (Park et al. 2003) for a more detailed introduction of the Grid Method itself.

Grid-based methods are commonly used in for example modeling heat transfer and fluid flow. Common for the problems for which this method is applied is that the problem boundary conditions are known, and the interior conditions are unknown. The process then involves splitting the area to be solved for into smaller unit grids, and permeate the boundary conditions into inner regions by applying governing equations to these unit grids successively. Each unit grid uses information from its own local boundaries. When applied to optimizing kinematic design in a TOD process, each unit grid represents the properties of one joint for one specific task point. The 4 design variables used for each joint are the Cartesian x, y, and z position and twist angle α, together denoted as $\mathbf{x}$. Each joint uses information from the previous and next joint in the kinematic chain and the same joint for the previous and next task point. Thus each joint can be treated separately, with no need for calculating the inverse kinematics of the full kinematic chain. This also means that these 4 properties are the only design variables for the optimization process.

For each joint a grid operation is performed. This includes first converting the design variables to DH parameters for the current joint, then evaluating the DH parameters with a weighted unit grid cost function. The DH parameters of each joint can be calculated based on the known position and twist angle of a joint and the previous and next joint in the kinematic chain, see (Park et al. 2003) for a detailed explanation. The grid operation is performed successively on all the joints in the kinematic chain and for all the task points used. A global convergence criteria based on the sum of the cost function for each grid operation determines when a sufficiently optimized global solution has been found.

The benefit of a fixed number of design variables is that the algorithm does not become less efficient as more DOF and task points are added. In this study this was an important factor, given that 12 different robots with 5-7 DOF were optimized for about 5 task points each. However, the Grid Method as originally conceived also has its limitations. The optimization process is only performed on one joint at a time, as this is the definition of the unit grid used. This makes the symmetry of two joints difficult to enforce, and symmetry is typically required for a climbing robot like the one designed for here. Take the 2nd joint in a 5 DOF robot like ASIBOT as an example, see Fig. 3.2. This robot is symmetric about the center 3rd joint. Not only should the fixed DH parameters for this joint be similar across task points, they should also be similar to the ones for the 4th joint. In a grid-based approach only local information is typically used, here information about the previous (1st) and next joint (3rd) in the kinematic chain, and the same joint in the previous and next task point. Even with information about the 4th joint it would be difficult to achieve symmetry, as the 4th joint is solved for at a later stage in the optimization process. A modified version of the Grid Method was developed to overcome this, which is described in the next section.

3.3.2.2 Modified Approach

The first modification to the Grid Method was to expand the unit grid to a pair of symmetric joints. This allows for enforcing the desired symmetry within each grid operation. Thus symmetric grid operations in general have 8 design variables, independent of the number of DOF and task points used. Fig. 3.3 depicts all the n joints (open circles) for the m task points to solve for. The notation used in the rest of this chapter is i for a given joint for task point j, while c denotes the center joint. The global base (closed circle, light grey) and the global task point (closed circle, dark grey) are fixed and act as boundary conditions. As can be seen indicated in the figure the symmetric unit grid uses information from its local neighborhood. In other words, each symmetric pair of joints knows the parameters of the next and previous joints in the kinematic chain as well as for the same joints in the next and previous task point.

The second change was to solve the problem as two manipulators, each with one of the global

end-effectors as base. The manipulators were named *left* and *right*, see Fig. 3.3. These *left* and *right* manipulators share the global center joint, and the local task point for each is the second to last joint for the other. In the original Grid Method the end-effector orientation required an additional constraint function as only the position was determined by the Grid Method. This was done by comparing the X and Z axis of the actual orientation of the end-effector with the desired task point orientation. By splitting the robot in two identical manipulators, both the base and the end-effector orientation of the robot are now defined. The problem then reduced to enforcing the orientation of the z-axis of the two manipulators' end-joints, to ensure the two manipulators are aligned properly. See section 3.3.2.7 for a description of this constraint.

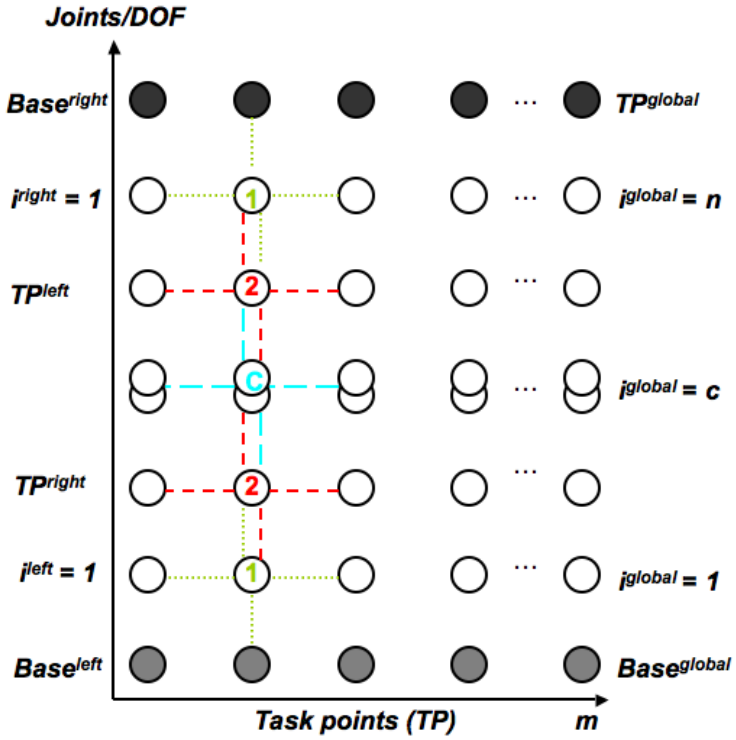


Fig. 3.3: *Symmetric unit grid visualization for a 5 DOF symmetric robot. The problem has 3 symmetric joint unit grids (denoted 1, 2 and c) for each task point.*

A final change in the original approach was needed to ensure that the first and last links are symmetrical. This is not possible with the original formulation, as a zero-length first link (link 0) was assumed, with the zero frame (at joint 1) and the base frame overlapping. The base frame is here instead defined to be at the end-effector of the first link. This can also be seen in Fig. 3.3.

3.3.2.3 Very Fast Simulated Annealing

Simulated annealing is an optimization technique typically applied to find a global optimal solution for problems with large search spaces. It is widely utilized, partially for its ability to get out of local minima, but also because it is relatively simple to apply. It has its roots in earlier algorithms for statistical mechanics, and the name "annealing" comes from a process in metallurgy used to improve the properties of metals. This process involves heating up a metal to a temperature where the atoms are allowed to move freely, then performing a controlled cooling dictated by the properties sought from the material. The goal is typically to increase the size of

the crystals and reduce the defects in the material. The simulated annealing works on a similar basis, with the solutions varying almost randomly at high temperatures, but slowly improves in the right direction as the temperature is decreased. The temperature is controlled by a an annealing schedule, which varies with the different versions of the technique. The technique used here is Very Fast Simulated Annealing (VFSA), as used in the original Grid Method (Park et al. 2006). It is a simplified version of the algorithm presented by (Ingber 1989). This technique's main improvement over the original is the improved convergence rate, which leads to shorter solution times. This is primarily done through a modified annealing schedule.

3.3.2.4 Symmetry Unit Grid Cost Function

As the symmetry unit grid has been expanded to two symmetric joints, the corresponding cost function also includes both, as can be seen in Equation (3.1) ($left$ and $right$ superscripts). The symmetric unit grid cost function consists of six weighted constraint functions. The equalization constraint (EC) ensures that the DH parameters of a joint are as close as possible to that of the same joint for the next and the previous task point (except joint angles for revolute joints). The limit constraint (LC) allows for constraining one or more of the DH parameters for a joint to within a given range. The obstacle avoidance constraint (OA) penalizes the depth of penetration of a joint or link with a spherical obstacle. These three constraints are unchanged from the original Grid Method and are presented in detail in Park, Chang and Yang (Park et al. 2003). The symmetry constraint (SYM), total length constraint (TLC) and center constraint (CEN) are described in section 3.3.2.5, 3.3.2.6 and 3.3.2.7, respectively.

$$\begin{aligned} F_{SUG}(\mathbf{x}_{i,j}^{left}, \mathbf{x}_{i,j}^{right}) \;=\; & w_{SYM} \cdot f_{SYM((i,j)^{left},(i,j)^{right})} \\ + \; & w_{TLC} \cdot f_{TLC((i,j)^{left},(i,j)^{right})} \\ + \; & w_{CEN} \cdot f_{CEN((i,j)^{left},(i,j)^{right})} \\ + \; & w_{EC} \cdot (f_{EC(i,j)^{left}} + f_{EC(i,j)^{right}}) \\ + \; & w_{LC} \cdot (f_{LC(i,j)^{left}} + f_{LC(i,j)^{right}}) \\ + \; & w_{OA} \cdot (f_{OA(i,j)^{left}} + f_{OA(i,j)^{right}}). \end{aligned} \tag{3.1}$$

3.3.2.5 Symmetry Constraint

The symmetry constraint function is used to ensure that the two symmetric joints in the unit grid are similar. As can be seen from Equation (3.2), the constraint applies to both the previous $(i-1)$ and the following (i) link with respect to the joint in question. Refer to Fig. 3.1. A special weight ω_{ang} is applied to the angular values to equalize the order of magnitude of angles with those of distances. In addition the link twist angles are either added or subtracted depending on the definition of the DH parameters for the symmetric robot. This is governed by the n number of k_i values, which are set to 1 or minus 1 respectively.

$$\begin{aligned} f_{SYM((i,j)^{left},(i,j)^{right})} \;=\; & f_{LS((i-1,j)^{left},(i-1,j)^{right})} \\ + \; & f_{LS((i,j)^{left},(i,j)^{right})} \\ + \; & w_{ang} \cdot f_{\alpha S((i-1,j)^{left},(i-1,j)^{right})}, \end{aligned}$$

$$where:$$

$$f_{LS((k,j)^{left},(k,j)^{right})} = (l_{k,j}^{left} - l_{k,j}^{right})^2 + (d_{k,j}^{left} - d_{k,j}^{right})^2, \tag{3.2}$$

$$and:$$

$$f_{\alpha S((r,j)^{left},(r,j)^{right})} = (\alpha_{r,j}^{left} + k_i \cdot \alpha_{r,j}^{right})^2.$$

3.3.2.6 Total Length Constraint

For a climbing assistive robot the total length is important, as it influences among other the portability of the robot and the structural requirements of the docking stations. The limit constraint (LC) only applies to the DH parameters of the one joint being optimized, therefore a new total length constraint (TLC) was introduced. The total length of the robot, L_{total}, was simply defined as the sum of all the link lengths and offsets for the complete robot. The constraint function penalizes a total length larger than a set maximum or smaller than a set minimum, as can be seen in Equation (3.3).

$$f_{TLC} = \begin{cases} (L_{total} - L_{max})^2 & for\ L_{total} > L_{max} \\ (L_{total} - L_{min})^2 & for\ L_{total} < L_{min} \\ 0 & otherwise. \end{cases} \tag{3.3}$$

3.3.2.7 Center Constraint

The two symmetric manipulators share the last joint which thus have the same x, y, z position and twist angle α. However, for the *left* and *right* last joints to be equivalent, the joint Z axes for each must also coincide. This was ensured with an additional center joint constraint, which can be seen in Equation (3.4). Here $\mathbf{z}$ represents the Z unit vector of the final joint of the respective manipulator. The constraint was applied to all the joints in the kinematic chain to take into account their possible effect on the orientation of the global center joint. Forward kinematics was used to propagate these effects from a given joint to the center joint.

$$f_{CEN((i,j)^{left},(i,j)^{right})} = ||\mathbf{z}_{c,j}^{left} - \mathbf{z}_{c,j}^{right}||^2 \tag{3.4}$$

3.3.2.8 Obstacle Definition

The obstacles in the environment are important constraints when performing tasks with a robotic manipulator, and should therefore be included in the optimization. The obstacle avoidance constraint used here is the same as that used in (Park et al. 2003). The environment obstacles were represented by simple spheres. For each iteration the distance to all spherical obstacles is calculated for all links and all joint origins. If the distance from a link or joint origin to the center of a sphere is less than the radius of the sphere a collision is defined. The cost of the collision is then proportional to the depth of the penetration. To be able to model a more realistic environment, the obstacles used for the study presented here were initially defined as simple rectangular prism primitives. These obstacles were then grown by the radius of the robot, here 62.5 mm was used. This was done to approximate the physical space available to the actual robot. Finally, each grown obstacle were approximated by a set of overlapping spheres. These spheres were used as the basis for the obstacle avoidance constraint described above. The obstacles used in this study can be seen in Fig. 3.6 and an example of an obstacle represented as grown spheres can be seen in Fig. 3.8.

3.3.2.9 Global Convergence

The Grid Method performs an optimization for each unit grid (here a symmetric joint pair), starting from the extremes, for each task point. The global convergence criterion used is then the sum of the cost functions for all the symmetric unit grids, across m task points and c symmetric joint pairs. The criterion can be seen in Equation (3.5).

$$F_{total} = \sum_{j=1}^{m} \left(\sum_{i=1}^{c} F_{SUG}(\mathbf{x}_{i,j}^{left}, \mathbf{x}_{i,j}^{right}) \right) < u. \tag{3.5}$$

3.3.2.10 Special Formulation for Spherical Center Joint

Some robot configurations, like the 7 DOF robot used for the study presented here, have joints with several intersecting axes. This also means that a large number of DH parameters need to be forced to zero, restricting the search for solutions. As was done in (Park et al. 2003) with a spherical wrist, the three central intersecting joints for the 7 DOF configuration used here were replaced with a spherical joint located in the center pitch joint location. These joints could then be represented as a single joint with a single set of design variables $(x, y, z, \alpha)_c$. The three joint angles were then solved for using inverse kinematics during optimization of this joint.

3.3.3 Practical Considerations

3.3.3.1 Setting the Constraint Function Weights

All the constraint functions used here have zero as the optimum. The angles and distances for the kinematic description all have magnitudes on the order of 10^0 as radians and meters are used as units. This simplifies the process of setting the weights manually somewhat. A strategy used here was to set the obstacle avoidance weight several orders of magnitude higher than the remaining weights, to reduce the chance of the robot getting trapped in a local minima with a collision. For example in the overlap between two spheres representing an obstacle. The remaining weights were then manually tuned to attempt to balance the respective terms in the cost function. In (Park et al. 2006) an adaptive method was used to alter the weights during execution. This can help reduce the time and effort to find an optimal solution, and is particularly important if measures that are not simple constraints are included. For example the dexterity measure used in the original Grid Method, that should be minimized, but not necessarily to zero. Adapting the weights in a similar manner for the modified Grid Method is beyond the scope of the work presented here, but should be feasible given the similar structure of the constraint functions used.

3.3.3.2 Efficiency of the Modified Grid Method

A simple comparative study was performed to assess the effect of the symmetry-specific modifications made to the Grid Method. The modified method (*ModGrid*) was compared with the original Grid Method (*Grid*) and the General Formulation Method (*GFM*). The latter was also used as the basis for comparison in (Park et al. 2003), and performs the optimization over all the joints and task points at the same time. It is thus a non-grid approach and has $4*m*n$ design variables, as compared with 4 and 8 for the original and modified Grid Method, respectively. The total computation time in Matlab on a 3 MHz Intel Core 2 Duo was compared for 5 and 7 DOF robots optimized for 2, 3 and 4 task points. The constraints used were EC and C (with weights 1 and 2, respectively). For *Grid* and *GFM* the center constraint C was replaced with an equivalent of the Desired Orientation Constraint (DOC) used in the original Grid Method (with weight 1). The convergence criteria used was increased with both the number of task points and DOF used, specifically $u = 0.0005 * m * n$. As can be seen in Fig. 3.4, the results indicate that while not as efficient as the original, the modifications made to enable optimization of symmetric robots still makes it more than an order of magnitude faster than the *GFM* as the number of DOF and task points increases.

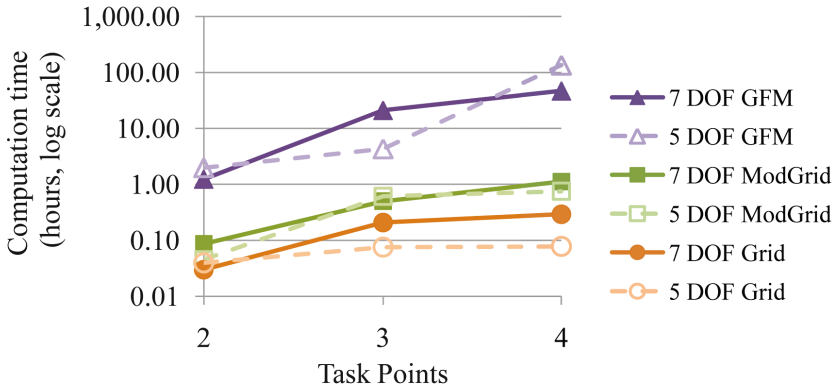


Fig. 3.4: *Total computation time for the original (*Grid*) and modified Grid Method (*ModGrid*), as well as the General Formulation Method (*GFM*).*

3.3.4 Design Methodology

3.3.4.1 Overview

This section describes the application of the modified Grid Method for symmetric robots to the design of an assistive climbing robot. As described in the introduction this type of robot has a range of additional requirements in comparison to most robot designs. One of the most important is that the robot is intended to be operated by, and in close proximity to, disabled users in their own homes. The task-oriented approach underlying the Grid Method suites this well, as it allows the designers to take specific tasks and environments into account in the design of the kinematic structure. For this study a kitchen environment was used, based on an exact model of the real kitchen environment in the RoboticsLab at Universidad Carlos III de Madrid (UC3M) (Jardon et al. 2008), see Fig. 1.3a. For this environment four scenarios were envisioned. Each scenario is centered around a specific area or appliance in the kitchen, and includes possible docking station locations, obstacles and task points. The four scenarios, *cabinet*, *refrigerator*, *dishwasher* and the user's *wheelchair* are all described in more detail in section 3.3.4.2. For each design scenario a standard design process was followed, as seen in Fig. 3.5.

First, all likely task points and docking station locations were defined for the scenario. This was based on previous experience with the ASIBOT robot and previously recorded user preferences for tasks (Balaguer et al. 2005). For the scenarios presented here 25 (*dishwasher*) to 72 (*cabinet*) task points were defined. Several potential docking station locations were also defined. Ideally the robot would be optimized to be able to perform tasks for all docking stations in each scenario. However, to limit the number of task points for the optimization algorithm, it was chosen to first identify one "optimal" docking station for each scenario from which the tasks were assumed to be performed. This was done by assessing the number of tasks feasible from each docking station location with the original ASIBOT kinematic structure.

The Grid Method has been shown to be effective for both high numbers of DOF and for reasonable numbers of task points, 7 task points were for example used in (Park et al. 2006) for a 6 DOF robot. In the study presented here several different robot configurations were to be optimized and only standard performance desktop computers were used. A subset of around 5 task points was therefore used for each optimization. With such a subset of tasks and an "optimal" docking station 3 robot configurations (with 5, 6 and 7 DOF) were optimized for each scenario. The three configurations chosen are further described in section 3.3.4.3. The constraints used and parameters optimized are detailed in section 3.3.4.4.

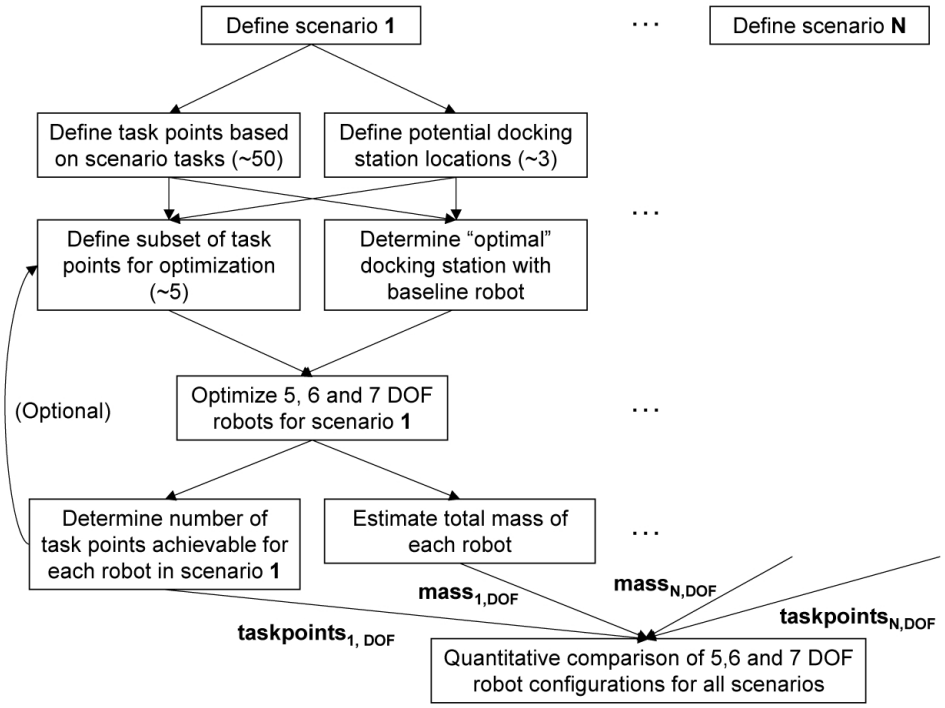


Fig. 3.5: *Design methodology proposed for climbing assistive robots.*

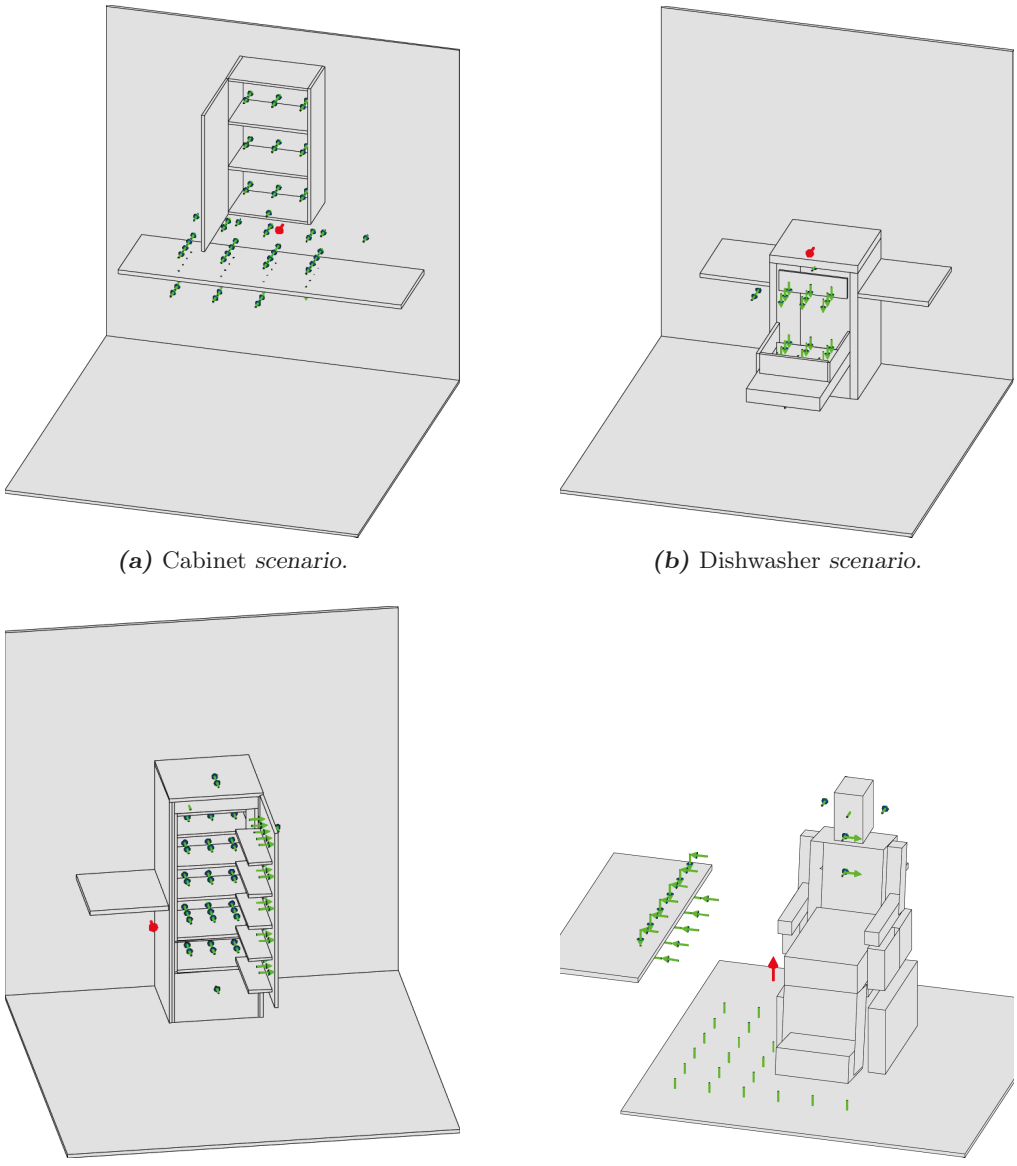
Once the 12 designs (4 scenarios with 3 robots each) had been optimized, the next step was to try to quantify the advantages and disadvantages of each design. The first criteria chosen was to quantify the number of task points reachable with each robot design in each scenario. This would help evaluate the optimization procedure as well as establish a global optimal design across the four scenarios. This was performed in a virtual environment, and is further explained in section 3.3.4.5. In addition the total mass of each robot design was estimated, which is described in section 3.3.4.6.

3.3.4.2 Task Scenarios

One of the most important aspects of performing a task-oriented design is that the robot can be made to guarantee that certain desired tasks will be achievable. This should then allow for the successful completion also of similar tasks. The ASIBOT robot is designed to perform a range of tasks from fixed docking stations and from the user's wheelchair. Therefore 4 different scenarios were selected, each with a set of desirable tasks and potential docking station locations. The 4 scenarios can be seen in Fig. 3.6. As mentioned earlier, one docking station was chosen as the most suitable to perform tasks in each scenario. Additional docking station locations were then included as tasks, to make sure the robot would have mobility to perform tasks in the given scenario, but also to transfer outside it.

Two main types of task points were defined. The first required all 6 DOF (position and rotations) of the end-effector to match the desired task. For example docking station locations, as the docking process in the current ASIBOT (and likely also in future versions) require the robot to insert the end-effector with a high degree of accuracy in both pitch and yaw and then perform a pure roll movement to lock into position. Other task points had a relaxed yaw requirement (in the end-effector frame), enabling the 5 DOF robot to perform tasks where the yaw rotation was not essential. For example when picking up an object like a bottle from the kitchen desktop. For

the docking stations defined as tasks another task point was also added 100 mm offset normal to the docking station itself, to represent the required movement for performing a docking. The different tools used when performing different tasks were also taken into account in the task point placement. For example the task points representing the user eating from a spoon. These were placed 100 mm in front of the users mouth to simulate the length of a typical spoon held in the end-effector of the robot.



(a) Cabinet *scenario.*

(b) Dishwasher *scenario.*

(c) Refrigerator *scenario.*

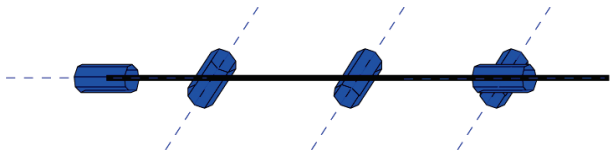
(d) Wheelchair *scenario.*

Fig. 3.6: *Task scenarios defined. Smaller green arrows indicate the origin and direction of the Z axis of the end-effector for a given task point, while larger red arrows indicate the docking station location used as the base for the scenario shown.*

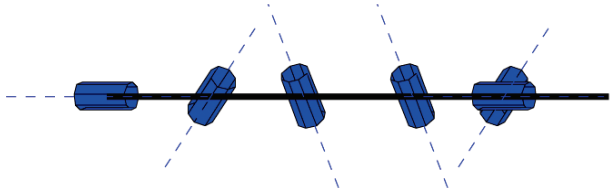
3.3.4.3 Robot Configurations

The next step in the proposed methodology was to choose a subset of robot configurations that seemed suitable for the scenarios selected. By robot configuration is here meant the definition

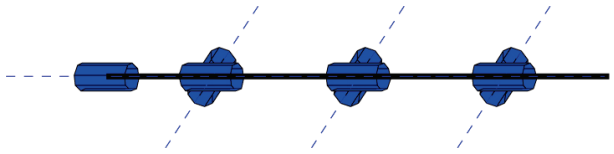
of a set of basic properties that were not to be optimized. This included the number of DOF and the twist angle of each joint. First, the robot should have 5, 6 or 7 revolute DOF. The current design has 5, limiting it in some tasks where 6 DOF of the end-effector is required. A 6 DOF robot may be able to perform these tasks, while a 7 DOF in addition may add redundancy. Second, the robot should have the ability to roll the end-effectors about the local z-axis. This may be needed to perform the docking procedure and to interchange end-effectors. Third, the robot should be symmetric. The definition of robot symmetry used here is that the robot must be able to perform the same set of tasks when docked with the left and when docked with the right end-effector in the same docking stations.



(a) *5 DOF configuration.*



(b) *6 DOF configuration.*



(c) *7 DOF configuration.*

Fig. 3.7: *Robot configurations used as the basis for the optimization process.*

With these constraints 3 robot configurations were defined. The 5 DOF version was based on the current ASIBOT configuration, as can be seen in Fig. 3.7a. The design has a roll joint in each end, and three pitch joints in between (*RPPPR*). Symmetry is across the central joint (joint 3). The 6 DOF robot was made symmetric about the center link to comply with the requirements for symmetry. The end joints had to be roll joints, according to the constraints applied. For the symmetric joint pair adjoining the center link yaw joints were chosen. The remaining pair of joints was then selected to be pitch joints, giving *RPYYPR*, as seen in Fig. 3.7b. The 7 DOF configuration was based again on the ASIBOT design, but with roll joints added to the two links adjoining the central pitch joint (*RPRPRPR*). See Fig. 3.7c. The properties to optimize for each configuration were the two pairs of symmetric link lengths (for the 6 DOF configuration also a central link). In addition the joint limits and any required joint offsets needed to accommodate the range of movement could be defined for each design.

3.3.4.4 Kinematic Optimization of Robot Designs

The modified Grid Method described in section 3.3.2 was used to obtain one kinematic design for each DOF configuration for each scenario. An example can be seen in figure Fig. 3.8, showing a 7 DOF robot optimized for 5 task points in the *refrigerator* scenario. The spheres represents the obstacles seen in Fig. 3.6c. A common set of constraints were used for all optimizations. The first was that the maximum length should not be longer than 1500 mm, to make sure the

robot could still be used on the user's wheelchair and remain portable. The second constraint used was to limit the first real link (from the end-effector to the first non-roll joint) to between 200 and 400 mm. Less than 200 would be very difficult mechanically as it houses both a roll joint and a docking/end-effector mechanism. The remaining length of the robot was constrained to be within 400 mm and 1000 mm. This consisted of one symmetrical pair of links for 5 and 7 DOF, each of between 200 mm and 500 mm. For 6 DOF one symmetrical pair of links of between 100 and 250 mm as well as one central link of between 200 and 500 mm. Thus the minimum feasible distance between joints was assumed to be 100 mm. These constraints were also made with the requirements for housing the electronics in mind. A static 7th joint was added to the middle of the central link of the 6 DOF configuration when optimized. The DH parameters of this joint were forced to zero during optimization, except for the link length. This enable the optimization of robots symmetric about a central link instead of a central joint, without changes in the algorithm used.

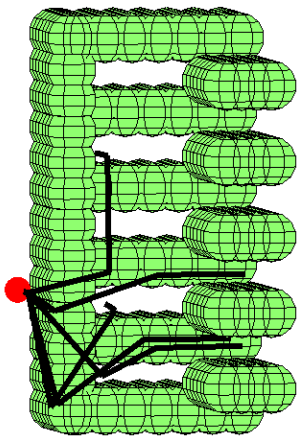


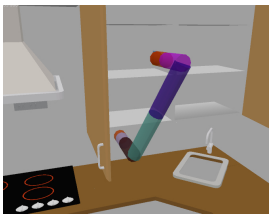
Fig. 3.8: *Example 7 DOF robot optimized for the* refrigerator *scenario. Poses indicate task points used for the optimization. Actual obstacle used for the optimization process for this scenario.*

For the joint limits, ASIBOT was used as a reference. The roll joints were given a 360° range. The pitch joints in ASIBOT all have ±135° joint limits, mainly due to lack of joint offsets. A 5 or 7 DOF robot with a joint offset in the central pitch joint could potentially have an increased workspace, especially close to the base of the robot. Such a robot could also satisfy the 4 constraints described in section 3.3.4.3, including that for symmetry. This should be further explored in future studies. The weights of the unit grid cost function were set manually, and could be varied for each optimization. The goal was to achieve a smallest possible design that could reach the set of task points given without exceeding joint limits or colliding with the obstacles used. A satisfactory design was assumed when this could be achieved with a reasonably coherent design for all task points used (typically ±10 mm for each link).

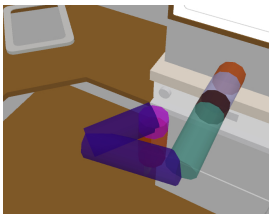
3.3.4.5 Virtual Testing of Robot Designs

Virtual testing was used to compare the kinematic designs obtained for each configuration from each scenario to obtain a global optimal kinematic design. The criteria used was the number of task points achievable in each case. The testing was performed in a simulation based on the Marilou Robotics Studio environment, with realistic models of the obstacles in all scenarios and simplified robot models representing the designs obtained. Examples of robots being tested in

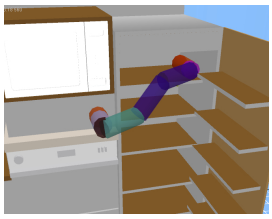
the four scenarios can be seen in Fig. 3.9. Collisions between the robot and the environment were simulated and joint limits used to limit the workspace of the robots. Approximately the same docking station location as used for the optimization was used, but the testing was performed on the complete set of task points for each scenario. For each task point, the robot was commanded to the desired Cartesian position and orientation of the task. If allowed for the specific task (see section 3.3.4.2), the yaw rotation requirement (in end-effector frame) for the task was ignored in the differential inverse kinematics solver used. If the robot successfully reached the task point (with the rotation requirement met according to the task type) a success was recorded. If the robot was prevented from reaching the task point due to joint limits or collisions with the environment other poses, if available, were explored. For example trying an "elbow-up" pose instead of "elbow-down", or using the inherent redundancy in the 7 DOF designs.



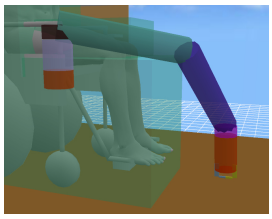
(a) C5 *robot in the cabinet scenario.*



(b) D6 *robot in the dishwasher scenario.*



(c) R6 *robot in the refrigerator scenario.*



(d) W7 *robot in the wheelchair scenario.*

Fig. 3.9: *Virtual testing of the robot designs.*

3.3.4.6 Mass Estimation of Robot Designs

Another aspect considered when comparing the robot designs found was the expected total mass of the robot. As the designs are relatively similar to the original ASIBOT design in terms of DOF, total length and payload requirements, this robot was used as the baseline for the estimations. The main assumption made was that the robot designs would have a similar distribution of mass across subsystems as that in ASIBOT. This meant that the total mass, M_{total}, could be split into two parts. One part varies with the number of DOF and actuators, denoted as M_a. This includes the mass of the motors, reductors and motor drivers and was found to represent about 40% of the total mass in ASIBOT (Jardon 2006). The remaining 60% then represents the mass of the structure, common electronics and the end-effectors docking mechanisms, denoted as M_s. Both masses were assumed to vary linearly with the total length of the robot. This allowed for a crude estimation of the additional structural and motor torque requirements for lifting the same payload with a longer moment arm. The calculation of the mass for each robot can be seen in Equation (3.6), where L signifies the total length of the design and n the number of DOF. The ASIBOT data used was a mass of 12 kg and total length of 1336 mm.

$$
\begin{aligned}
&M_{total} = M_a + M_s, \\
&where: \\
&M_a = 0.4 \cdot M_{ASIBOT} \cdot \frac{L}{L_{ASIBOT}} \cdot \frac{n}{n_{ASIBOT}}, \\
&and: \\
&M_s = 0.6 \cdot M_{ASIBOT} \cdot \frac{L}{L_{ASIBOT}}.
\end{aligned}
\tag{3.6}
$$

Tab. 3.2: *Optimized robot designs with lengths and estimated mass (in mm and kg respectively)*

DOF	Robot	$L_{extreme}$	L_{middle}	L_6	L_{total}	M_{total}
5	*C5*	385	365	n/a	1500	13.5
	D5	200	295	n/a	990	8.9
	R5	195	315	n/a	1020	9.2
	W5	260	400	n/a	1320	11.9
6	*C6*	260	235	340	1330	12.9
	D6	215	220	300	1170	11.3
	R6	200	220	230	1070	10.4
	W6	250	215	380	1310	12.7
7	*C7*	270	415	n/a	1370	14.3
	D7	210	320	n/a	1060	11.0
	R7	255	340	n/a	1190	12.4
	W7	205	465	n/a	1340	14.0

3.4 Results

3.4.1 Optimized Kinematic Designs

The results of the optimization process can be seen in Table 3.2. The robots are named with a single letter (*C-cabinet*, *D-dishwasher*, *R-refrigerator* and *W-wheelchair*) followed by the number of DOF. Displayed are the different optimized lengths for each robot. $L_{extreme}$ is the length of the symmetric link pair from the end-effector to the first non-roll joint ($MU+DS$ in the original ASIBOT design in Fig. 3.2), L_{middle} is for the other symmetric link pair (equivalent to BR in Fig. 3.2), while L_6 is the total length of the center link unique to the 6 DOF configuration. The total length, L_{total} and the total estimated mass, M_{total} are also shown for each robot.

As can be seen from the table, the robots optimized for *cabinet* has the longest $L_{extreme}$ in comparison with those optimized for other scenarios. From the docking station used this makes sense, as the robot is required to avoid the underside of the cabinet to access any of the shelves. With a large distance from the base at the wall to the first non-roll joint the obstacle is easier to avoid. The extra length in the symmetric section of the robot near the end-effector does not negatively affect the tasks in the cabinet (reaching positions 100 mm above the shelf and parallel to the shelf surface). The total length for the robots optimized ranges from 990 mm (*D5*) to the maximum 1500 mm (*C5*). The robots optimized for the *cabinet* scenario are the longest, while the robots optimized for the *refrigerator* and *dishwasher* scenarios are the shortest. This reflects the tasks present in each scenario, with most task points in the two latter being easily accessible by the robot, as they are in close proximity to the docking station and without a significant obstacle in-between. The average length of the robot designs found was 1223 mm and the average estimated mass was 11.9 kg (ranging from 8.9 kg to 14.3 kg).

3.4.2 Quantitative Comparison of Robot Designs

Table 3.3 shows the results of the virtual testing of each robot design for each of the four scenarios. For each case a number from 0 to 1 is shown, representing the ratio of the task points achieved to the task points available in the given scenario. The result for a robot both optimized and tested for a given scenario (for example *C5* for the *cabinet* scenario) is highlighted in bold.

Tab. 3.3: *Comparison of tasks achievable with each robot design in each scenario (C-Cabinet, D-Dishwasher, R-Refrigerator, W-Wheelchair).*

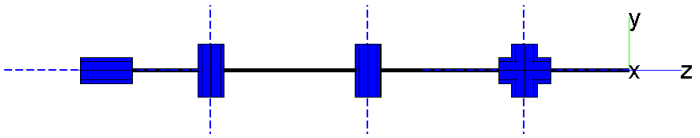
DOF	Robot	*C*	*D*	*R*	*W*	Total
5	*C5*	**0.57**	0.56	0.51	0.38	2.01
	D5	0.39	**0.56**	0.37	0.28	1.60
	R5	0.44	0.56	**0.46**	0.34	1.81
	W5	0.54	0.56	0.55	**0.61**	2.26
6	*C6*	**0.26**	0.80	0.52	0.44	2.03
	D6	0.31	**0.80**	0.40	0.43	1.93
	R6	0.28	0.80	**0.52**	0.30	1.90
	W6	0.26	0.72	0.43	**0.49**	1.91
7	*C7*	**0.74**	0.56	0.54	0.79	2.62
	D7	0.63	**0.56**	0.48	0.46	2.12
	R7	0.65	0.56	**0.58**	0.57	2.37
	W7	0.64	0.56	0.54	**0.90**	2.64

As can be seen from the table, 10 out of 12 of the robots optimized for a given scenario also had the best results for this scenario and number of DOF. For the two that did not, the scores were not far from the average of the other designs with the same DOF for that scenario. This was the *C6* robot with a score of 0.26 for the *cabinet* scenario (average 0.28 for 6 DOF designs) and the *R5* robot with a score of 0.46 in the *refrigerator* scenario (average 0.47 for 5 DOF designs). An unpaired t-test found no significant difference between the mean score of the 12 robots optimized and tested in the same scenario (0.59) and the remaining robots (0.50), $t(46) = 1.66$, $p = 0.103$. However, this is partially due to the very uniform results of the *dishwasher* scenario in general. For the remaining three scenarios a weakly statistically significant increase of 23% was found, $t(34) = 1.96$, $p = 0.059$. Paired t-tests were used to explore the level of generalization provided by the robots optimized for given scenarios. It was found that the mean score for the robots optimized for the *cabinet* scenario (0.56) was significantly higher than for those optimized for the *dishwasher* (0.47) and *refrigerator* (0.51) scenarios, with $t(11) = 2.76$, $p = 0.019$ and $t(11) = 2.24$, $p = 0.047$, respectively. There was a weakly statistically significant increase for the robots optimized for the *wheelchair* (0.57) with respect to the *dishwasher* scenario, $t(11) = 2.09$, $p = 0.061$.

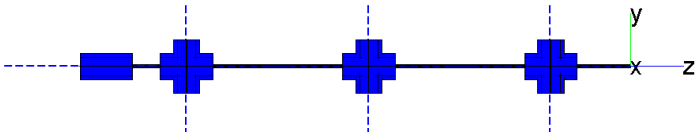
A limitation of the approach taken is that only a subset of the potential task points identified was used in the optimization process. As a result not all the tasks existing in a given scenario were possible with the "optimal" robot. However, it is not clear how else to deal with a large set of real-world tasks, complex obstacles and tight robot constraints under the TOD approach. At some point a subset of tasks that are achievable with the design constraints used must be selected for the optimization process to converge. The design approach followed here is made less sensitive to the selection of this subset by the following quantitative comparison of the different robot designs on the full set of tasks in all the scenarios. For designers this means that quantitative trade-offs can also be made when selecting the final design, for example based on user-preferences about the importance of one scenario versus another.

The scores obtained for each design were summed up to get a total score. The 7 DOF robot

W7 achieved the highest total score, 2.64, and is compared to the original ASIBOT design in Fig. 3.10. The average for 7 DOF designs, 2.44, was also higher than the two other configurations. Interestingly the average for the 5 DOF configuration (1.92) was about the same as that for 6 DOF (1.94). This reflects one of the issues with the 6 DOF configuration seen during testing, the very different poses required for a given task point as compared to the 5 and 7 DOF arms. One example can be seen in Fig. 3.11, comparing a 5 DOF and 6 DOF robot with a very similar size on the same task in the *cabinet* scenario. The task point shown is to reach another docking station location underneath the overhead cabinet. As can be seen from the figure the 6 DOF version required a very different pose from the 5 DOF causing it to hit the 135° joint limit of the last pitch joint for a seemingly simple task. A similar issue was seen in task points where the robot was required to enter a constrained space, for example picking up objects from the shelves in the *cabinet* and *refrigerator* scenarios. The geometry of the 5 and 7 DOF configuration resembles that of the human arm. As the environments in assistive robotics are typically designed for the latter this may give the 5 and 7 DOF an advantage. This can also be of importance in the ability of the user to effectively use the robot in cluttered environments. If the robot pose for a given desired end-effector position and orientation is difficult to imagine the user may have trouble predicting and reacting in time to avoid collisions during direct control of the robot. Automatic obstacle avoidance is an active research field in robotics, however there are additional safety concerns related to operating near a disabled person. A degree of user involvement in controlling this type of robot is therefore usually desirable.



(a) *Original ASIBOT design, length 1336 mm.*



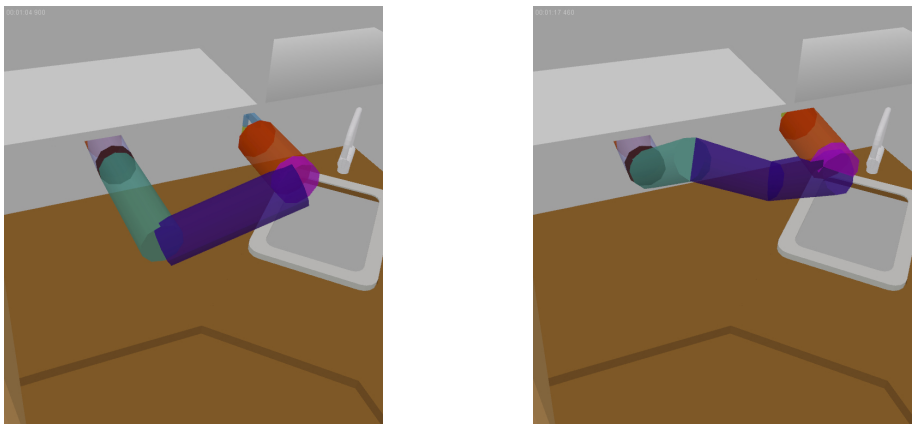
(b) *Design with highest score (W7), length 1340 mm.*

Fig. 3.10: *Comparison of original and "best" design, drawn to scale.*

The design most similar to the current ASIBOT robot, *W5*, gained the fourth highest score (2.26). This may indicate that the current design approaches the optimal for the DOF and the tasks used. It also indicates that moving to a 7 DOF design could increase the performance by about 17%, but that this would have to be traded off with a 17-18% increase in mass. However, the use of more light-weight materials than what was available for the original ASIBOT would likely reduce such an increase in mass somewhat. The results are also dependent on the docking station locations and tasks chosen. Here the docking station used for the testing in each scenario was chosen based on the best performance of the 5 DOF ASIBOT robot. However, the results obtained are at least indicative of the comparative performance of the different designs, given the wide range of tasks and obstacles used.

3.5 Discussion and Conclusion

In this chapter a task-oriented design process was applied to the kinematic design of an assistive manipulator, ASIBOT. Such assistive robots are intended for a comparatively wide range of tasks in a user's home, often in close proximity to a disabled person. For example



(a) R5 *robot design.* ***(b)*** R6 *robot design.*

Fig. 3.11: *Comparison of 5 and 6 DOF configuration for the same task point. The task represents reaching an alternative docking station location mounted underneath the overhanging cabinet.*

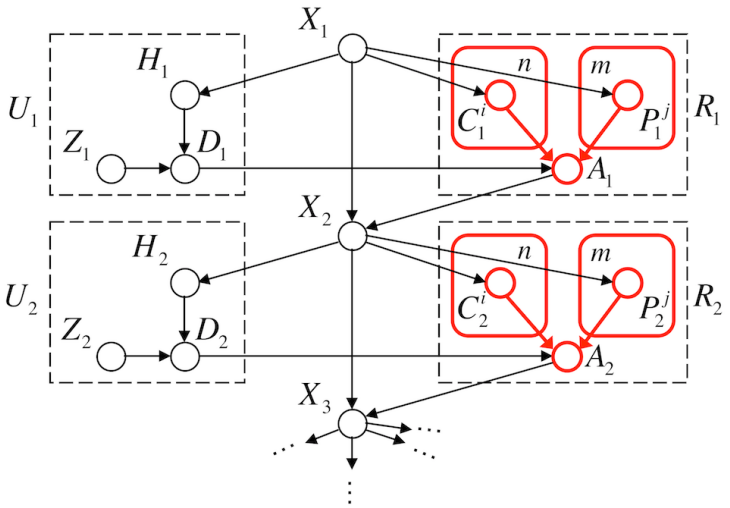
allowing the user to pick up, fill and drink from a glass, a task that may not have been possible without personal assistance before. The task-oriented approach ensures the design process is centered around the intended tasks and the end-user.

Climbing robots have other particular properties, including symmetry, high mobility in the environment and a low total mass. A design methodology was proposed to address these issues. A definition of 4 typical kitchen scenarios was made, including task points of interest and obstacles. 12 different robot designs were then optimized on a subset of tasks, each based on one of the 4 scenarios and with 5, 6 or 7 DOF. The optimization method used was based on the Grid Method, which allows for an efficient optimization even for a large number of DOF and task points. A modified version of the method was developed to allow for the optimization of symmetric robots.

A quantitative comparison of the designs was then used to decide upon a suitable robot kinematic structure. This involved estimating the total mass for each design, based on previous experience with ASIBOT. In addition a comparison of the achievable tasks for each design was performed in a realistic simulation. The results showed that a 7 DOF design could increase the number of tasks achievable by 17% in comparison to the best 5 DOF design, but that this would come at a cost of a 17-18% increase in total mass. The use of the methodology proposed ensured that this type of quantitative trade-offs could be identified and evaluated based on the actual performance of the designs on all the tasks desired. Although a good design can also be reached without such an approach, the methodology proposed should increase the likelihood of finding one for a wide range of tasks and environments.

Future improvements to the work presented in this chapter could include an automated adjustment of the weights for the constraint functions (which are here manually tuned), to simplify the use of the algorithm. Furthermore, there is a potential for improving the convergence characteristics using for example a stepsize adjustment, which gradually reduces the region that is searched for an optimum as the number of iterations increase. For a new version of ASIBOT, the inclusion of joint offsets, and thus potentially increased joint ranges, could be included in the study. Finally, a trade-off between the number of docking stations required and the size of the robot could be of interest.

4. LEARNING TO ASSIST THE USER IN LIMITING COLLISIONS



4.1 Introduction

As was mentioned in Section 1.2.4, there exists several approaches for enabling the user and the robot to share their sensing, control and planning capabilities. Such approaches are also known as shared control. There is for example considerable work on mobile robots that have an adjustable degree of autonomy. Here shared control is on the lower end of the scale of potential autonomous modes. See for example (Goodrich 2001). It is not necessarily straightforward to extend these techniques from mobile platforms to 5-7 DOF manipulators however, due to the increase in complexity and computational burden. As an example, if the joint angles are used as the state there can be an exponential increase in the size of the state-space with DOF. For assistive manipulators there is also much less work available on shared control. Some approaches are aimed at a higher-level interaction through speech or 2D interfaces, combined with some robot autonomy on specific tasks. This is also denoted as shared autonomy, and has recently shown considerable promise (Chen et al. 2013). This includes collaborative selection among known objects in the environment (Pitzer et al. 2011), and visual object selection by the user followed by visual servoing by the manipulator (Tsui et al. 2008). Exact models of the environment and their relation to every part of the robot can be hard to obtain and maintain. Reducing the reliance on such internal representations may therefore be desirable, especially if operating in partially unstructured and dynamic environments. It also seems beneficial to centre any such approach on the user controlling the manipulator, which brings us back to the speed-accuracy trade-off from Section 2.3.

4.1.1 Back to the Speed-Accuracy Trade-Off

The work presented in this chapter was initially driven by one idea: To provide the user of an assistive manipulator with aid in performing the speed-accuracy trade-off. Specifically, to let the robot help enforce the accuracy requirements of a task, indirectly allowing the operator to move with greater speed, and reducing the effect of his/her disability. This is not an entirely new idea, it has for example been shown that proximity-based force-feedback can improve performance on simple corridor-following tasks (Dennerlein et al. 2000). Reducing the velocity before an impact for a teleoperated robot manipulator has been explored in (Everett & Dubey 1998), with a single range finder mounted on the end-effector. The authors commented that:

"When it could be assured that the impact velocity would be limited, the operator could move the master much faster throughout the approach and contact task and thereby reduce the approach time".

This type of behaviour also seems plausible for the more general tasks attempted here. In effect enabling the user to move unconstrained on the gross movements, far from obstacles, while being aided when getting close to the target (or obstacles in the environment).

4.1.2 Sensing the Environment

The question then becomes how to help the user limit collisions through the robot's sensors. Ideally the sensing used should have a high probability of detecting an obstacle if indeed there is one, and a low rate of "false alarms" (a low probability of falsely detecting an obstacle). This is important for giving the user reason to trust, and agree with, the interventions performed by the system. One approach is to use an explicit model of the environment, the robot and the relative pose of the two, built with sensor data. As the environment is assumed to have both dynamic and unknown elements, a Simultaneous Localization and Mapping (SLAM) approach would likely be required. However, errors in sensor measurements, sensor to robot base, and robot base to robot end-effector also tend to add up. Robot-external sensors also suffer from occlusions, for example by the manipulator itself.

A set of distributed proximity sensors have the natural benefit of an increase in resolution with decrease in distance to obstacles, and a low amount of information to process in general. Both ultrasonic and infrared proximity sensors have previously been applied extensively in mobile robotics. The latter have also been recently used for online grasp adjustments (Hsiao et al. 2009). Earlier seminal work by Vladimir Lumelsky's group demonstrated full-body proximity sensing on industrial robot manipulators (Cheung & Lumelsky 1989). See Fig. 4.1, full videos are available at: https://directory.engr.wisc.edu/me/faculty/lumelsky_vladimir. Here over 1000 infrared proximity sensors were used to perform online movement planning and execution in unknown and dynamic environments. This remains a very challenging task today, even with the excellent sensors technology (e.g. time-of-flight and 3D laser sensors) and high-power computers available.

There are potentially many ways in which a distributed set of proximity and collisions sensors could be utilized. The goal should be to attempt to improve the performance of the system for most tasks that the user is likely to encounter, while not inhibiting the execution of any of them. It may be desirable to attempt to automatically adapt the usage to both operator abilities and scenario of usage, given the large quantity of parameters associated with a set of proximity sensors that have different physical location and potential usefulness for a given task. To maximize the predictability of the system from the user's perspective this adaptation should ideally occur in real-time, limiting the abruptness of changes in input-output mappings.

(a) *Unknown environment with humans.*

(b) *Dynamic environment: Ballet dancer.*

Fig. 4.1: *Examples of Lumelsky's work on full-body proximity sensing (Cheung & Lumelsky 1989). On the order of 1000 infrared proximity sensors were here distributed over an industrial manipulator, and used to plan (in real-time) and safely execute movements in unknown and dynamic environments.*

4.1.3 Distributed Adaptive Control (DAC)

One approach of interest is the Distributed Adaptive Control (DAC) paradigm (Verschure et al. 1992), (Verschure et al. 2003), (Pfeifer & Bongard 2006). The approach was inspired by a widely studied concept in behavioral sciences, Classical Conditioning (Pavlov 1927/1960). Briefly explained, this involves a Unconditioned Reflex (UR) like salivation in a dog when presented with food, the food here representing an Unconditioned Stimulus (US), and a Conditioned Reflex (CR) that is gradually obtained through the simultaneous (and often repetitive) presentation of a Conditioned Stimulus (CS) with the US (or shortly before). The CS can be for example the sound of a bell, as used in some of Pavlov's original experiments. After conditioning the sound of the bell (CS) could then be shown to trigger salivation in the dog, without the presence of food.

Although Pavlov's interpretation of such behavioral patterns is likely outdated, as was for example recognized by Ashby already in the 50's (Ashby 1957), the experimental results are still of importance and can be used to motivate work on learning in artificial systems. The DAC paradigm makes two key assumptions that are highly related to Classical Conditioning: 1. A predefined value system, expressed in combinations of US and UR. 2. A mechanism for associating CS representations to US representations. See Fig. 4.2 for a simplified example of DAC, a NN where a collision avoidance behavior is gradually learned in a mobile robot by associating proximity sensor readings (the CS) with collisions sensors (the US). The collision sensors are hardwired to predefined motor actions (the reflexes) that turn away from the obstacles, and the learning only occurs when there is a collision with the environment.

When the robot has a collision, the synapses between collision and proximity sensors that are active at that point in time are strengthened. These same proximity sensor will after several such collisions begin to activate the motor behaviors by themselves before the robot collides with the obstacle. Thus the system's "memory" is distributed over the learned connections (synapse weights), where the pattern of learned connections depends on the robot embodiment and the robot's interaction with its environment. We can say that the system therefore creates a sense-associate-act coupling, where the environment is used as a communications channel. It is also an example of both perceptual and behavioral learning through self-organization. In fact, in such a highly coupled sensorimotor system the structuring of the behavior of an agent through learning will also bias the future sensory information it receives. This can be taken advantage of to stabilize the system (Verschure et al. 2003). In the above example, once the robot no longer

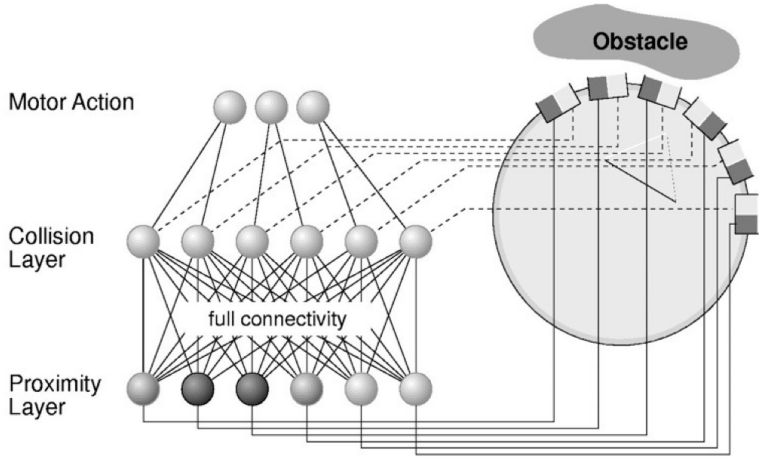


Fig. 4.2: *An example of a simple Distributed Adaptive Control (DAC) architecture for a mobile robot with proximity and collision sensors (Pfeifer & Bongard 2006).*

collides with the obstacles in its environment, it will stop learning new connections.

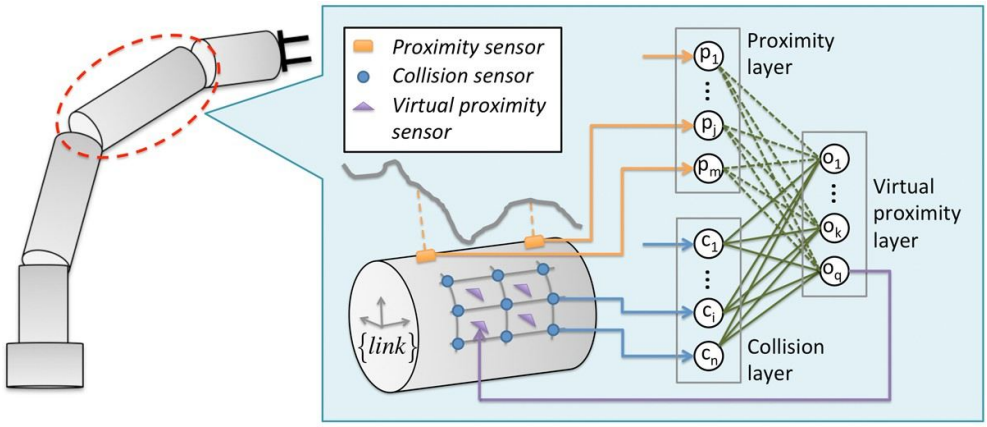
4.2 The Adaptive Collision-Limitation Behavior

An adaptive collision-limitation behavior for assistive manipulators was designed. The behaviour was based on a simple neural mechanism, inspired by the DAC approach, to adaptively limit the velocity of the robot before having collisions. This section describes the behavior in general, in terms of the sensing assumed, the algorithm used for slowing the robot down, and the NNs used for adapting the behavior. Several specific implementations of the behavior, and their experimental validations, will be discussed in the following sections (and the next chapter).

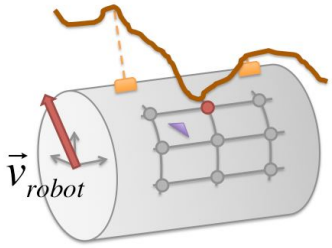
The behavior uses collision and proximity sensors distributed over the manipulator, see Fig. 4.3 for a graphical illustration of the concept. The approach assumes n collision sensors and m proximity sensors for each link, each of which is represented by a neuron in a respective input layer of a NN. An output layer with q linearly activated neurons is used to represent a set of virtual proximity sensors. The collision sensor neurons are hardwired to the virtual proximity sensor neurons (solid green lines in Fig. 4.3a), with the distribution of the weights depending on the proximity of the virtual proximity sensor to a given collision sensor. In the simplest case (the one used for the experiments presented here), each collision sensor has a unity weight connecting it to the closest virtual proximity sensor. Whenever a collision sensor activates, it thus also activated a virtual proximity neuron. The discounted Hebbian learning in Equation (4.1) is then used to associate this activation with the activation of the proximity sensors on the same link at the time of a collision. See Fig. 4.3b. This association occurs by increasing the respective weights $w_{k,j}$ between real and virtual proximity sensors (dashed green lines in Fig. 4.3a).

$$\Delta w_{k,j} = \frac{\gamma}{m}(\eta o_k p_j - \epsilon w_{k,j}). \tag{4.1}$$

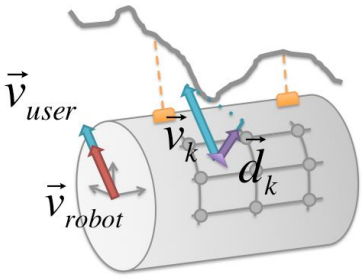
Each of the q virtual proximity sensors have a distance vector $\vec{d}$ associated, the magnitude of which depend on the learned associations and the activation of the physical proximity (and collision) sensors. Given that the NN deals with proximity, the magnitude of $\vec{d_k}$ was defined as the inverse of the activation of the output neuron for the same sensor (o_k). The collision-limitation behavior then reduces the magnitude of the commanded velocity at each instant (from $\vec{v}_{user}$ to $\vec{v}_{robot}$) depending on the maximum proximity ratio r over all links. This is shown graphically



(a) *The link-specific Neural Network (NN) for adapting the usage of the proximity sensors during collisions with the environment. NN weights in green: dashed lines indicate Hebbian learning (proximity layer), while solid lines indicate fixed weights (collision layer).*



(b) *Association of activation in proximity sensors and collision sensed with the environment during initial user-commanded movements ($\vec{v}_{robot} = \vec{v}_{user}$).*



(c) *Reduction of user velocity ($||\vec{v}_{robot}|| < ||\vec{v}_{user}||$) using the projected displacement ($\vec{v}_k$) and learned distance vector ($\vec{d}_k$) of virtual proximity sensor k.*

Fig. 4.3: *The adaptive collision-limitation behavior for assistive manipulators.*

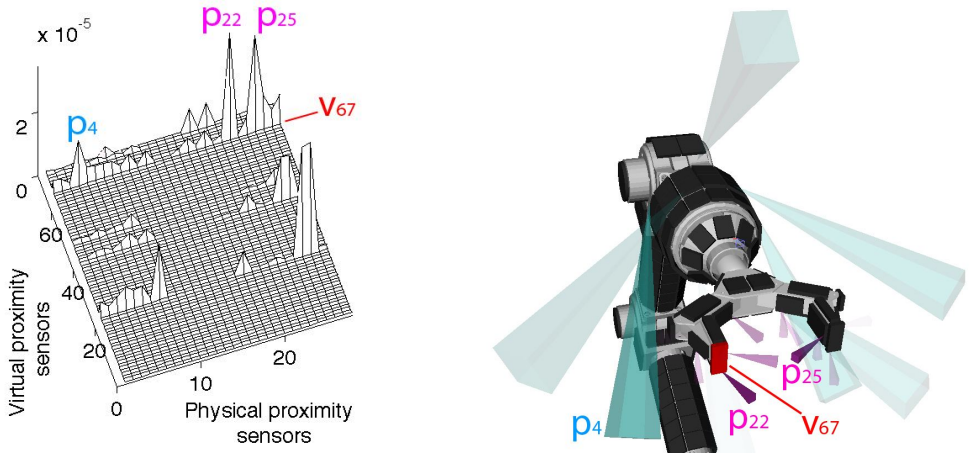
Algorithm 1 Reduction of the end-effector velocity commanded by the user, $\vec{v}_{user}$, based on the translational velocities of the virtual sensors, $\vec{v}_{trans,k}$, and the vectors representing the virtual sensors, $\vec{d}_k$, where $k = 1, \cdots, q$.

for $link = 1$ to N **do**
  for $k = 1$ to q **do**
    $proj_k = \vec{v}_{trans,k} \cdot \widehat{d}_k$
    $r_k = \frac{\alpha_{proj} + \beta_{proj} proj_k}{\|\vec{d}_k\|}$
  end for
  $r_{link} = \max_k(r_k)$
end for
$r_{robot} = \max_{link}(r_{link})$

$$\vec{v}_{robot} = \begin{cases} \vec{v}_{user} & if\ r_{robot} \leq 1, \\ \frac{\vec{v}_{user}}{r_{robot}} & otherwise. \end{cases}$$

in Fig. 4.3c. As can be seen in Algorithm 1, the proximity ratio for a given virtual proximity sensor k increases with proximity to an object (low $||\vec{d}_k||$) and with a high translational velocity in the direction of the virtual sensor vector (projection of $\vec{v}_k$ on $\widehat{d}_k$). The velocity commanded is

then reduced proportionally to the maximum proximity ratio (if larger than unity) over all links used.



(a) *Neural Network (NN) weights.* **(b)** *Corresponding visualization for* v_{67}.

Fig. 4.4: *Example Neural Network (NN) weights for final link (participant 5 in experiment). Visualization of weights for one virtual proximity sensor shown (v_{67}). Transparency of square pyramid representing the field of view of a given physical proximity sensor is made to vary with the corresponding weight connecting it to v_{67}. High transparency indicates low weight.*

Each virtual proximity sensor can thus be associated with multiple proximity sensors and fixed to multiple collision sensors, and the number can be scaled to fit the computational resources available. For the experiments performed here the virtual proximity sensors are assumed to be collocated and co-directed with the collision sensors (i.e. $q = n$ and $\widehat{d}$ normal to the collision sensor surface). See Fig. 4.4 for an example of the learned NN weights for one link.

4.3 Benchmarking on a Simplified Robot Embodiment

4.3.1 Introduction

For this first experiment a simplified model of an assistive robot was used, represented by the end-effector, see Fig. 4.5. This was done to mainly focus on the interaction between the user and the online adaptive shared control, rather than the issues related to sensorizing and limiting the movements of a full robot manipulator. That is, this first experiment was intended to establish wether a collision limitation behavior could effectively be adapted to a user without being unpredictable, and wether the performance could be increased on tasks where the speed-accuracy trade-off had to be performed by the user.

4.3.2 Implementation

The user could control the translational velocities of the end-effector in the end-effector frame, $\vec{v}_{robot} = [v_x, v_y, v_z]^T$. The proximity sensors were simulated as point distance detectors with a nominal range of 50 to 300 mm. For distances lower than 50 mm, 50 mm was returned. Infrared proximity sensors typically give ambiguous readings below the minimum distance. In a physical implementation the sensors would therefore have to be offset with respect to the zone of measurement, or be complemented by low-range sensing. Gaussian noise with a standard deviation of 10 mm was added to the readings. In all 20 proximity sensors were used, distributed

over the robot end-effector. 20 discrete collision sensors were simulated with the same sensors, with a range of 10 mm. In a physical implementation collision sensors based on physical contact would likely be used.

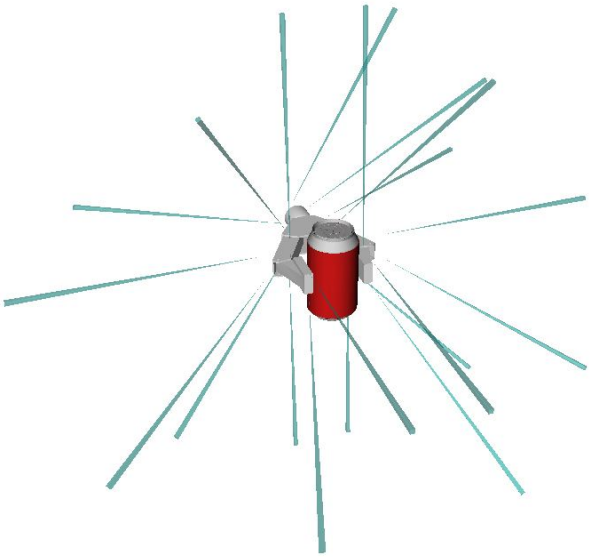


Fig. 4.5: *Robot embodiment and sensors used for the experiment. Ranges of simplified proximity sensors shown.*

4.3.3 Experiment Method

4.3.3.1 Participants

5 able-bodied participants were used, all university graduate students at UC3M. There were 2 female and 3 male, all right-handed. All had previous experience with 3D input devices. The mean age was 26.8, with a range from 26 to 27.

4.3.3.2 Simulated Environment and Tasks

The virtual experiment setting can be seen in Fig. 4.6. The experiment was performed in the OpenRAVE virtual environment (Diankov 2010), running at approx. 50 Hz. The 5 tasks performed consisted in moving a can from an initial position in front of the user to a specified target position in a kitchen cabinet or on a kitchen desktop. The user was given the view of a user in a wheelchair, see Fig. 4.6a. A transparent and enlarged target-can was used to indicate the position and tolerance (by the scale of the target) required to finish the task.

4.3.3.3 Simulated Disability

Noise was added to the user input, according to Equation (4.2). This was Gaussian noise, low-pass filtered at 2 Hz and generated independently for each Cartesian component of the noise vector ($\vec{z} = [z_x, z_y, z_z]^T$). The magnitude of the velocity caused by the noise increased proportionally to the magnitude of the translational velocities commanded by the user, with some noise existing also when the user did not indicate movement (non-zero α_{noise}). See Fig. 4.6b for example trajectories.

(a) *The simulated environment, as seen by the participants (on one task).*

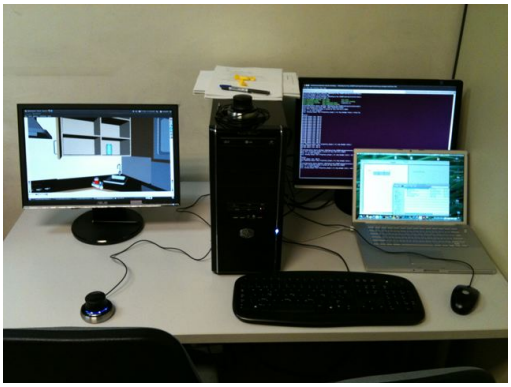
(b) *Example translational trajectories. Partic. 1, shared control.*

Fig. 4.6: *The virtual experiment setting used.*

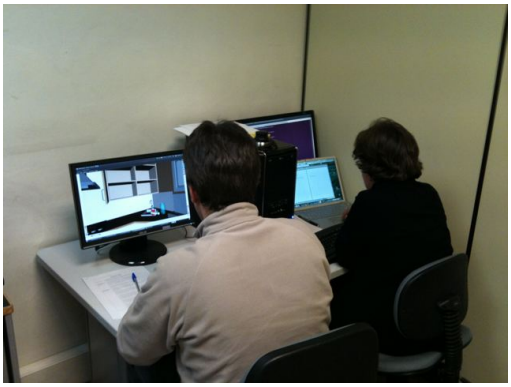
$$\begin{aligned} &\vec{v}_{user} = \vec{v}_{input} + \vec{v}_{noise}, \\ &where: \\ &\vec{v}_{noise} = \vec{z}(\alpha_{noise} + \beta_{noise}||\vec{v}_{input}||). \end{aligned} \tag{4.2}$$

4.3.3.4 Physical Setup

The physical experiment setup can be seen in Fig. 4.7. The input device used was a SpaceNavigator 6 DOF joystick. The simulation of the robot in the environment seen in Fig. 4.6a was displayed on a 20 inch computer monitor at a distance of about 1 meters. A colored timer was also shown.



(a) *The workstation used, with SpaceNavigator joystick in bottom left corner.*



(b) *A participant performing the experiment, with an experimenter on the right.*

Fig. 4.7: *The physical experiment setup used.*

4.3.3.5 Procedure

The testing was performed over one hour for each participant, and informed consent was obtained from each. First each participant was given 2 training sessions to get familiar with the experiment setup, followed by 6 main sessions. Each session had 25 trials per participant, with 5 repetitions of 5 different tasks. The 2 training sessions and the first 2 main sessions the

(a) *Without noise added.* ***(b)*** *With noise added.*

Fig. 4.8: *Trajectories from pilot study on the effect of the signal-dependent noise.*

shared control was not used. For the remaining 4 the system was activated. A timer was shown to the participants, which began running when the participant pressed an indicated button. If the hand itself (not the can) collided with the environment, the hand was reset to the initial position, while the timer kept running. The trials that had collisions were rerun. For all trials the participants were instructed to attempt to achieve the lowest times possible, while keeping in mind that collisions were costly in terms of time.

4.3.3.6 Data Collection and Conditioning

The experiment data was recorded at 50 Hz. The data used for calculating the mutual information was normalized and discretized to 10 states. The mutual information was then estimated by the histogram method. The InfoMeth Matlab toolbox (Lungarella et al. 2005) was used. The time series were the x, y and z Cartesian components of the commanded velocity ($\vec{v}_{robot}$ in Algorithm 1 and A in Fig. 2.3) and the noise velocity ($\vec{v}_{noise}$ in Eq. 4.2 and Z in Fig. 2.3), averaged in time over every 4 points recorded. All 5 successful attempts for all 5 tasks were used.

4.3.3.7 Preliminary Pilot Studies

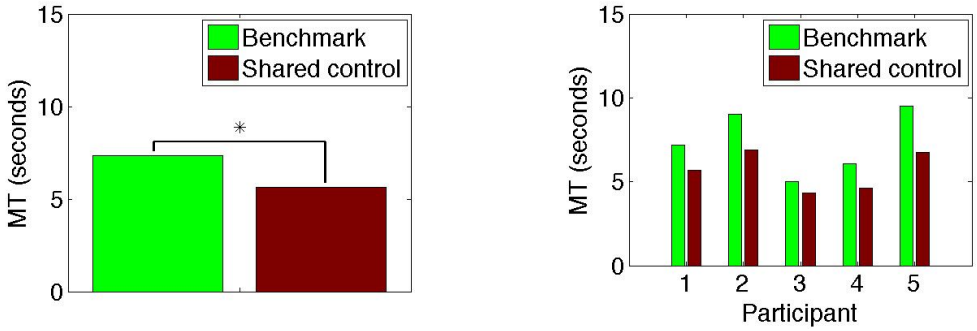
Two pilot studies were performed before the main experiment. The first was used to test the experiment setup and assess the effect of adding noise to the user input. Two participants were used. The recorded user trajectories for the five target positions can be seen in Fig. 4.8a. As can be seen, the participant performed the task quite consistently across repetitions. For a second condition the signal-dependent noise was added to the user input. The resulting trajectories can be seen in Fig. 4.8b, clearly showing the negative effect of the noise.

A second pilot study was performed to test the effect of a static collision limitation behavior on user performance. Three participants were used. Indications of improvements in performance were found, both for MT and the information measures from Section 2.2. There were two important lessons learned however. First, that the participants complained of a lack of control over the shared control, which lead to the development of the adaptive mechanism described above. And second, that it is very hard to control for the errors in such an experiment, which lead to the inclusion of the cost of collision in MT, as explained in section 2.3.5.

4.3.4 Results

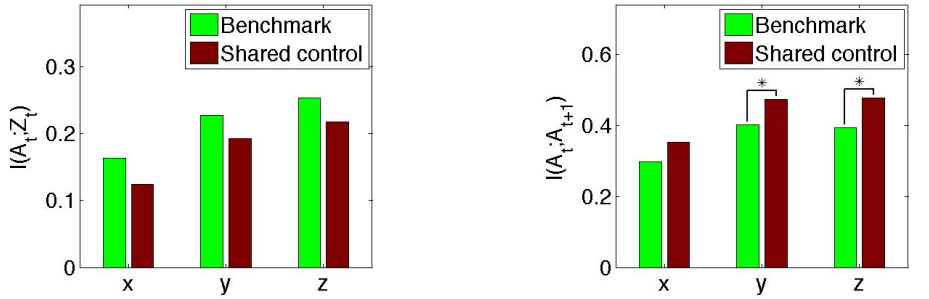
4.3.4.1 Mean Time with Collisions Included

The results for the MT can be seen in Fig. 4.9. All participants had a reduction in the metric with the shared control, ranging from a 13.2% (participant 3) to a 29.2% (participant



(a) Average MT over participants. *(b) MT for each participant.*

Fig. 4.9: *The Mean Time (MT) with and without (benchmark condition) the shared control.*



(a) Measure for controllability, the mutual information over the velocity components for the noise added (Z) and the robot actuation (A).

(b) Predictability of execution, the mutual information over velocity components for the robot actuation (A) at two points in time.

Fig. 4.10: *The results of the information theoretic metrics. Calculated over complete successful trajectories.*

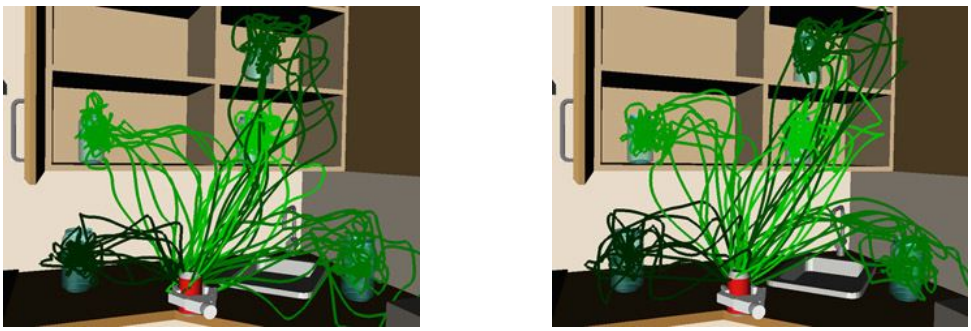
5) improvement. The 23.1% difference in the average metric over participants was statistically significant. A paired t-test was used, with $t(4) = 4.82$, $p = 0.009$.

4.3.4.2 Controllability from the User's Perspective

The results for the controllability from the user's perspective can be seen in Fig. 4.10a. There was a decrease in the mutual information, and thus increase in the controllability, for all velocity components with the shared control. Using a paired t-test no strictly significant differences were found on the .05 level, although the x component was weakly significant, with $t(4) = 2.63$, $p = 0.058$.

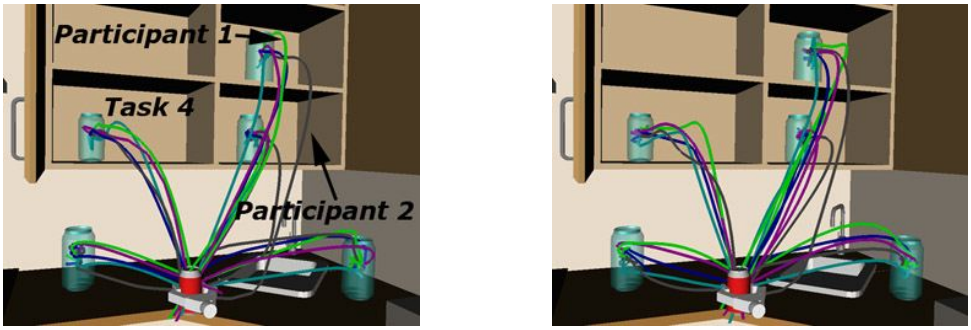
4.3.4.3 Predictability of Execution

The results for the predictability of execution can be seen in Fig. 4.10b. There was a increase in the predictive information for all velocity components with the shared control. The differences in the y ($t(4) = -6.65$, $p = 0.003$) and z ($t(4) = -4.07$, $p = 0.015$) velocity components were statistically significant.



(a) *Benchmark condition.* ***(b)*** *Shared control condition.*

Fig. 4.11: *Successful trajectories followed by participant 1. Color shading indicates task attempted.*



(a) *Benchmark condition.* ***(b)*** *Shared control condition.*

Fig. 4.12: *Gaussian Mixture Regressions (GMR) for the successful trajectories of 5 participants. Color indicates participant.*

4.3.4.4 3D Trajectories

Fig. 4.11 shows the successful trajectories followed by one participant (participant 1) on the benchmark condition (Fig. 4.11a), and the shared control condition (Fig. 4.11b). The effect of the noise added can be clearly seen in the irregularity of the trajectories. Fig. 4.12 shows the Gaussian Mixture Regressions (GMR) for the successful trajectories of the 5 participants for the same tasks and the same session-pairs. All 10 trajectories were used for each regression. The GMR is used here to give an idea of the general strategies used by each participant. The GMM-GMR v2.0 Matlab toolbox (Calinon 2009) was used, with 5 components in the Gaussian Mixture Model (GMM). The data was initialized by k-means clustering and the GMM was trained using the Expectation Maximization (EM) algorithm.

4.3.4.5 Velocity Profiles

Fig. 4.13 shows an example of the effect of the shared control system on the (hand-local) velocities. An overlap of the $\vec{v}_{robot}$ and $\vec{v}_{user}$ trajectories means no assistance is provided. From 1.8 seconds onwards the positive x velocities (towards the backside of the cabinet) are limited. See Fig. 4.12. From 2.1 seconds onwards the shared control system starts preventing movements in the positive y direction (roughly towards the left with respect to the participants view), where the cabinet door is. Negative z velocities are limited from 2.3 seconds, slowing the robot down in the direction of the cabinet shelf underneath the target. Around 2.5 seconds there is a sharp

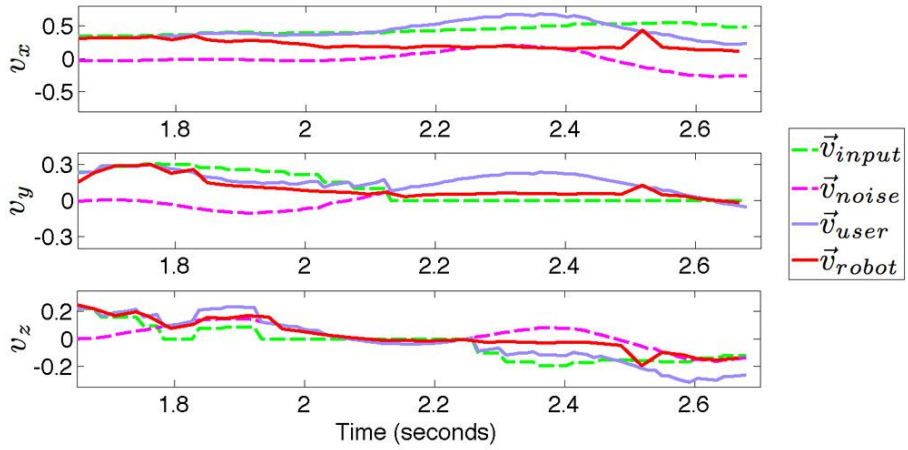


Fig. 4.13: *Cartesian x, y, and z components of the velocities (in hand frame) for the last second of one trajectory on task 4, session 8, for one participant.*

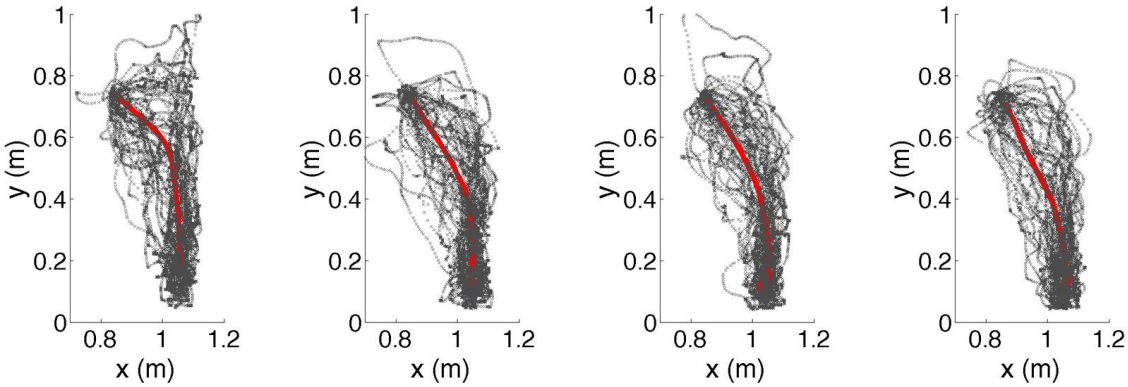
indent in the output of the shared control, likely the effect of a proximity sensor loosing sight of an obstacle temporarily. This is a limitation of the coarse distribution and the simplified models used for the proximity sensors.

4.3.5 Discussion

Although comparing performance over a longer time period would be beneficial, the results do give an indication of the potential for performance improvement with the shared control system. It should also be noted that the goal here is not zero MT. The above mentioned pilot studies showed that a no-noise condition had a 40% lower MT when compared to a condition with a similar noise as that used here. A reduction in the mutual information between the noise and the actuation with the shared control system seems to reflect the general trend of increased performance, and so does the increase in the predictability of execution. For both metrics, a larger population of participants would be needed to draw firm conclusions on the significance of the results however. So would a fully balanced experiment design. An interesting additional measure would be the subjective, or felt, controllability of the system from the user's perspective, and its potential correlation with the quantitative controllability measure used here.

The execution of the trajectories in Fig. 4.12 can be seen to be reasonably similar across participants, except for one participant (participant 2). Some changes in strategy can be seen between conditions. However, there is a limit to what can be deduced from these figures, given the lack of a "ground-truth" against which the trajectories can be compared. This highlights an important problem for benchmarking such systems. It is for example not given that the "optimal" trajectories followed by able-bodied participants will also be the "optimal" for participants with a (simulated) disability using a shared control. Thus metrics such as Mean Square Error (MSE) seem less applicable than metrics that focus on the ability to do work, and the ability of the system to reject noise.

The participants also adapted during the experiment. This can be inferred from the increase in performance, as the user would necessarily have to move at a higher velocity to reduce the time to complete a given task, but also from changes to the trajectories followed in space. Fig. 4.14 shows an overhead view of the trajectories for all 5 participants and the resulting GMR for task 4. In the initial sessions the strategy adopted by the participants were to first move into the direct sight line to the target, and later move along this vector. This likely reduced the need to trust the estimation of distance in the depth direction, and to avoid collisions with the



(a) *Training, benchmark condition.* **(b)** *Testing, benchmark condition.* **(c)** *Training, shared control condition.* **(d)** *Testing, shared control condition.*

Fig. 4.14: *Trajectories (grey markers) for the successful trajectories of 5 participants, and the corresponding GMRs (bold red lines). The task begins in the upper left corner.*

open door. For the later sessions however, the participants seem to increasingly choose a direct trajectory to the target.

4.4 Full-Body Collision-Limitation Behavior

4.4.1 Introduction

The previous section used a simple "floating" end-effector that does not fully correspond to an assistive manipulator. First of all, for a robot arm with multiple links, the user will typically be controlling the Cartesian velocities of the end-effector, but may have collisions on a link that has a very different Cartesian velocity at that point in time. Thus the kinematics of the full robot arm has to be taken into account to make the collision-limitation behavior coherent for all links. Second, the arm will in many cases occlude the view of the user when performing tasks, as assistive manipulators are typically mounted on a wheelchair in close proximity to the user. Here the collision-limitation behavior can potentially also provide assistance to the user. This section extends the collision-limitation behavior to a full multi-link manipulator based on the simulated ASIBOT robot.

4.4.2 Implementation

There were two major changes in the implementation for this experiment. The first change was to improve the simulation of the proximity sensors, taking into account the non-linear relationship between distance measured and voltage output. In addition the field of view of the sensors was made more realistic. The second main change was to extend the collision-limitation behavior to a full multi-link manipulator.

4.4.2.1 Simulating Proximity Sensors

The final implementation had 68 proximity sensors in total. See Fig. 4.15. 18 were simulated as Vishay TCND5000 (max. dist. 50 mm). These were all distributed over the end-effector. The remaining sensors were simulated as Sharp GP2D120 (max. dist. 400 mm). All proximity sensors had a simulated 10 ° field of view, represented in the simulation by a square 6 by 6 array of point distance measurements. The lowest of the 36 point distance measurements was used at

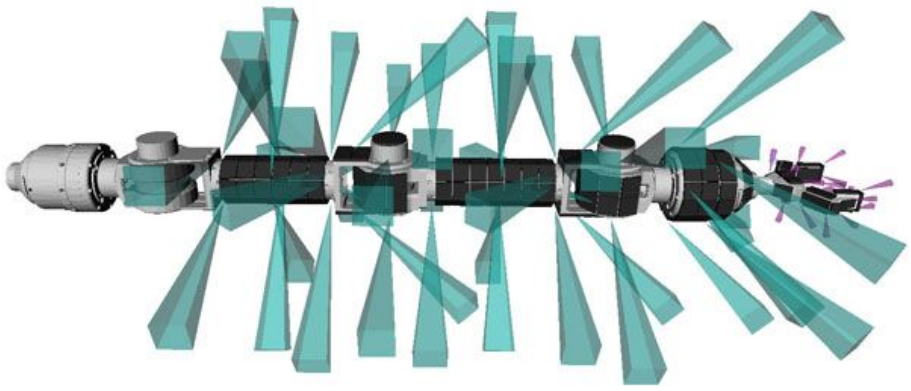
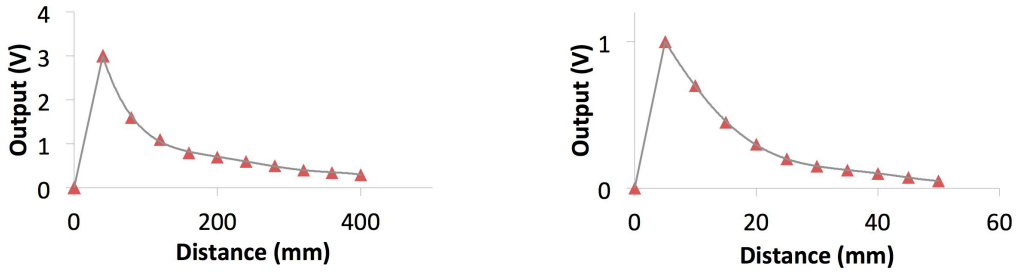


Fig. 4.15: *Collision sensors (black squares) and proximity sensors implemented on the virtual ASIBOT manipulator. Simulated field of view shown for each proximity sensor: Medium-range Sharp GP2D120 and short-range Vishay TCND5000 as green and purple square pyramids, respectively.*



(a) *Sharp GP2D120.* ***(b)*** *Vishay TCND5000.*

Fig. 4.16: *Plots of simulated voltage output for proximity sensors used. Gray line is output assumed, red triangles indicate calibration data points.*

any time. The voltage output of each proximity sensor was simulated based on the minimum distance measured, $prox_j$, and the calibration specifications seen in Fig. 4.16. This voltage was directly fed as input to the NNs (p_j). That is, the signal used by the NN varied inversely to the distance measured (in the nominal range of the sensor).

4.4.2.2 Extending to Multiple Links

Fig. 4.17 shows the schema for the collision-limitation behavior for a complete multi-link manipulator. The received velocities of the end-effector, $\vec{v}_{ee}$, are here represented in the robot base frame (b superscript). Using an iterative solver for the inverse Jacobian, the corresponding joint velocities for all joints are calculated. Then each link is treated separately. Using the known kinematic structure of the robot and the current joint angles, the translational velocities of each sensor for each link is calculated. These are then used together with the output of the link-specific NN to produce the maximum proximity ratio for that link, as described in section 4.2. The learning is thus spread over multiple instances of NNs, each running independently. Finally, the original commanded end-effector velocities are limited based on the maximum proximity ratio for the complete manipulator. The output velocity, $\vec{v}_{ee,out}$, is the user-commanded velocity $\vec{v}_{ee}$ divided by this ratio. The behavior will only activate if the ratio exceeds one. This enables the limitation of velocity based on the learned virtual sensor usage of the complete manipulator. Audio feedback was used to help the user assess when the collision-limitation behavior was

acting. This consisted of simple tones being played with breaks in between. The frequency of the alternation was proportional to the current maximum proximity ratio, see Fig. 4.17. The frequency of the tones was used to identify the link, from low frequency at the base to high frequency at the end-effector.

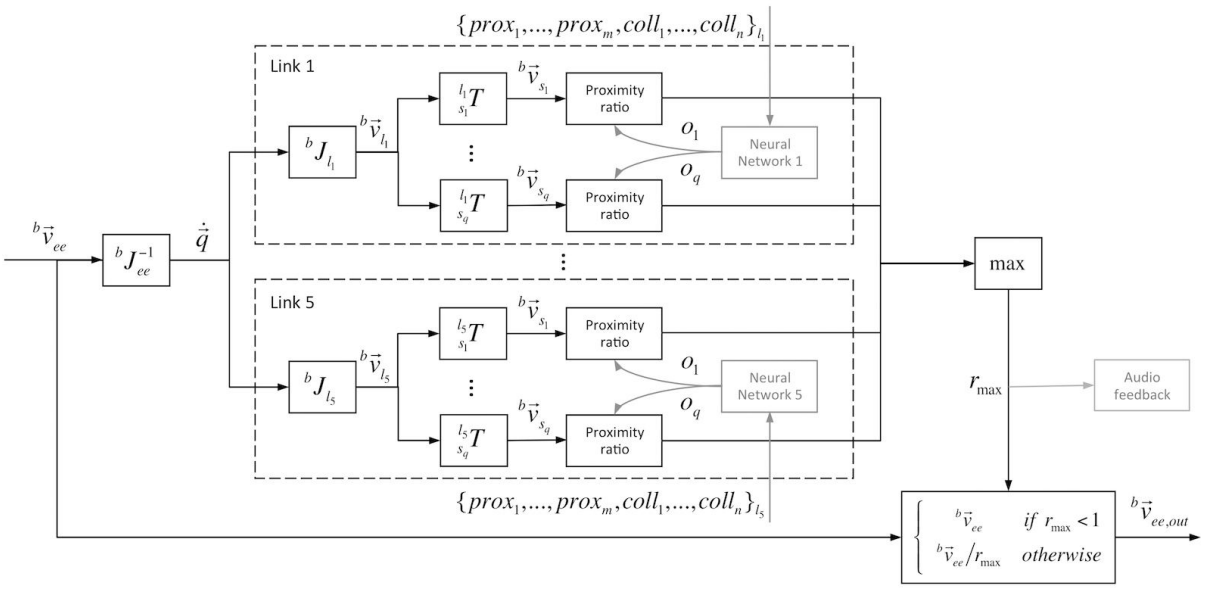


Fig. 4.17: *The full-body collision limitation schema. The current joint angles are used in calculating the Jacobian, but are here omitted for clarity. The b superscript is used to denote the robot base frame. Grey color indicates external modules.*

4.4.3 Experiment Method

4.4.3.1 Participants

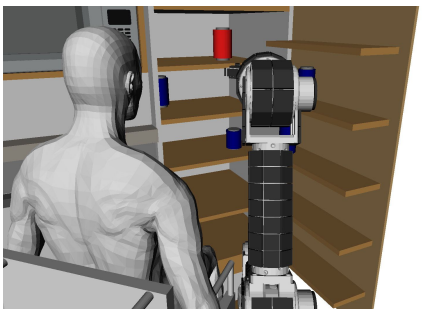
8 able-bodied participants were used, all university undergraduate and graduate students at UC3M. There were 3 female and 5 male, all right-handed. 4 had previous experience with 3D input devices. The mean age was 23.7, with a range from 19 to 40.

4.4.3.2 Simulated Environment and Tasks

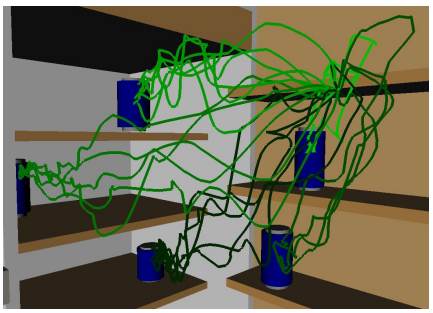
The simulated environment used in the experiment can be seen in Fig. 4.18. The ASIBOT robot is simulated to be attached to the right-hand side of the user's wheelchair. A view from behind the simulated user was given, to simulate the approximate size of the field of view that the participant would have sitting in the wheelchair. The tasks performed involved moving the end-effector of the robot from an initial resting position (see Fig. 4.18a) to a pre-grasp position around one of 5 simulated cans in the virtual environment. For a given trial the target can was red, while the remaining were blue. A trial was automatically judged as completed when the two fingers of the robot end-effector were positioned around the thickest part of the can, stopped or with a small remaining velocity magnitude. See Fig. 4.19 for an example execution on task 3.

4.4.3.3 Simulated Disability

Noise was added to the user input to simulate a generic physical disability, according to Eq. (4.3). This was Gaussian noise generated independently for each Cartesian component



(a) *The simulated environment, as seen by the participants. Robot in initial resting position.*



(b) *Example translational trajectories of end-effector. Participant 6, with shared control.*

Fig. 4.18: *The virtual experiment setting used.*

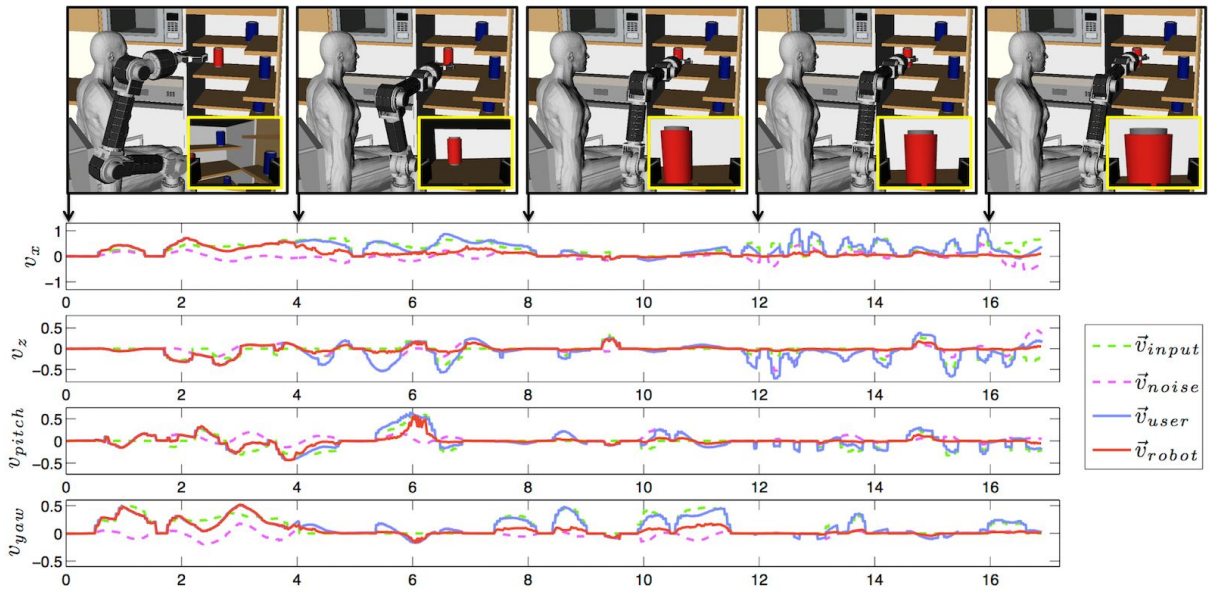


Fig. 4.19: *One attempt by participant 1 on task 3 with shared control. Cartesian x, z, pitch and yaw components of velocities (in end-effector frame) shown, with time in seconds on the x-axis. A discrepancy between the input ($\vec{v}_{user}$) and the output ($\vec{v}_{robot}$) velocities of the shared control means assistance is provided (from approx. 4 seconds onwards for this example). Actual robot poses along trajectory shown, but the camera angle is altered for visualization. Corresponding view from end-effector camera (as seen by the participant) shown in inserts. Description of phases: 0-4 seconds: highly coordinated gross movement, 4-8 seconds: adjustment of pitch during forward movement, 8-12 seconds: mainly yaw adjustments, 12-16 seconds: the final approach to the target.*

of the noise vector ($\vec{z} = [z_x, z_z, z_{pitch}, z_{yaw}]^T$), and low-pass filtered at 2 Hz. The magnitude of the translational velocities caused by the noise ($\vec{v}_{noise,trans}$) increased proportionally to the magnitude of the translational velocities commanded by the user ($\vec{v}_{input,trans}$), with some noise existing also when the user did not indicate movement (non-zero α_{noise}). Similarly for the rotational velocities ($\vec{v}_{noise,rot}$). See Fig. 4.18b for example trajectories.

$$\begin{aligned}
&\vec{v}_{user} = \vec{v}_{input} + \vec{v}_{noise}, \\
&where: \\
&\vec{v}_{noise,trans} = \vec{z}_{trans}(\alpha_{noise} + \beta_{noise}||\vec{v}_{input,trans}||), \\
&\vec{v}_{noise,rot} = \vec{z}_{rot}(\alpha_{noise} + \beta_{noise}||\vec{v}_{input,rot}||).
\end{aligned} \tag{4.3}$$

4.4.3.4 Physical Setup

The input device used was a SpaceNavigator 6 DOF joystick. The simulation of the robot in the environment seen in Fig. 4.18a was displayed on a 40 inch (approx. 102 cm) display (Samsung 3D TV, UE40D8000), at a distance of about 2 meters. The simulation was displayed in 3D and the participants used active 3D glasses ("Quad Buffer" stereo mode). This gave some perception of the depth of the scene. On a smaller display closer to the participant the simulated view from the end-effector camera was shown. See inserts in Fig. 4.19. A colored timer was also shown.

4.4.3.5 Procedure

The testing was performed over 2 days for each participant, with one hour committed per participant per day. Informed consent was obtained from each participant. The first day the participants were introduced to the experimental setup and was then given 3 sessions for training, followed by 2 sessions for establishing a benchmark. The second day the participants were introduced to the shared control, and were first given a maximum of 2 training sessions with the adaptation (learning) of the shared control activated. Then followed 2 sessions used to establish the performance with the (static) learned NN weights. Each session consisted of 3 repetitions of each of the 5 target can locations, for 15 trials in total. The arm was reset to the initial position if any part of the robot collided with the environment, the physical model of the user, or any of the target cans. For all trials the participants were instructed to attempt to achieve the lowest times possible, while keeping in mind that collisions were costly in terms of time.

4.4.3.6 Data Collection and Conditioning

The same approach as in section 4.3.3.6 was followed. However here the time series were the x, z, *pitch* and *yaw* Cartesian components of the commanded velocity ($\vec{v}_{robot}$ and A), the user input ($\vec{v}_{user}$ and H), and the noise velocity ($\vec{v}_{noise}$ in Eq. 4.3 and Z in Fig. 2.3). The data for participants 3 and 8 was not used for the time series analysis, given that each had suspected issues with the noise added for two sessions. For this experiment attempts with collisions were not repeated, and the time series analyses were therefore conducted over the first and last 6 seconds of the trajectories only (minimum time for completion was around 7 seconds). All 3 attempts for all 5 tasks were used.

4.4.4 Results

4.4.4.1 Adaptation

See Fig. 4.20 for examples of the development of the NN weights for the final link for three participants. A system with a static level of assistance could easily become another obstacle to overcome for the user. It can be seen that the adaptation was completed in about 20 minutes for each participant and that there were individual differences in the development. A corresponding visualisation of the final usage of the proximity sensors is given in Fig. 4.21. While participant 3

primarily received assistance when close to obstacles on the left, participants 5 and 6 had a more symmetric usage of the proximity sensors. The latter two participants also used the in-hand sensors, which are useful for slowing the robot down in the last moments of the tasks.

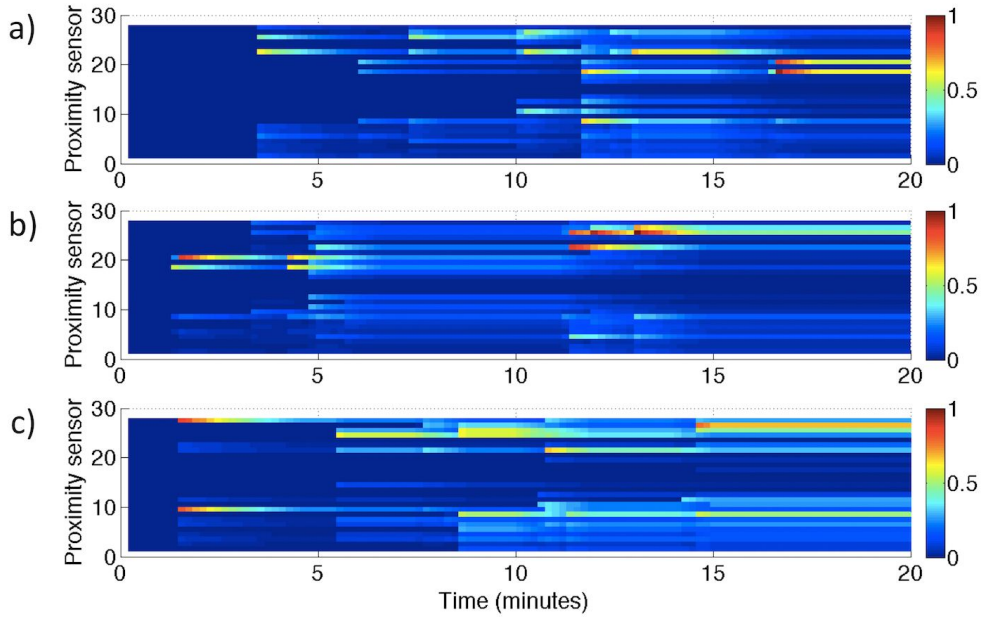
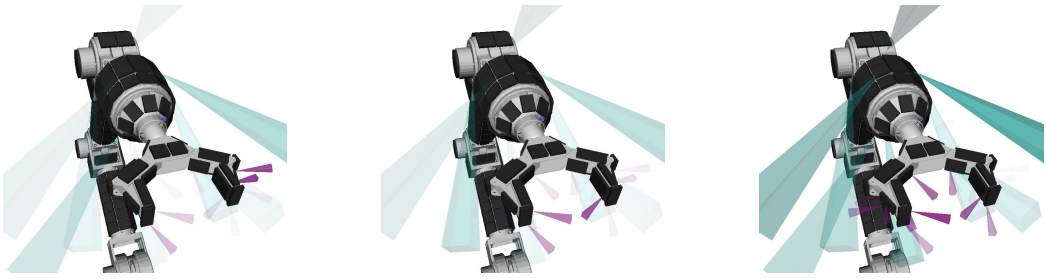


Fig. 4.20: *Examples of the development of the Neural Network (NN) weights for the final link. Mean weights for each proximity sensor of final link, normalized with maximum over 3 participants: a) 3, b) 5 and c) 6.*

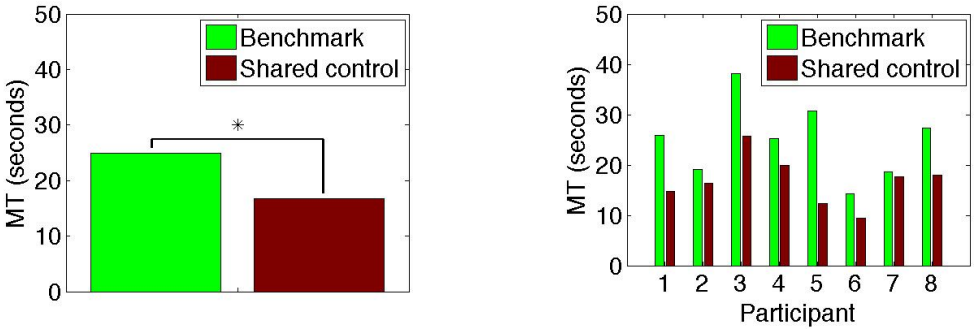
4.4.4.2 Mean Time with Collisions Included

The results for the MT for the experiment can be seen in Fig. 4.22. All participants had a reduction in the metric with the shared control, ranging from a 5.3% (participant 7) to a 59.9% (participant 5) improvement. The 32.5% difference in the average metric over participants was statistically significant. A paired t-test was used, with $t(7) = 3.96$, $p = 0.005$.



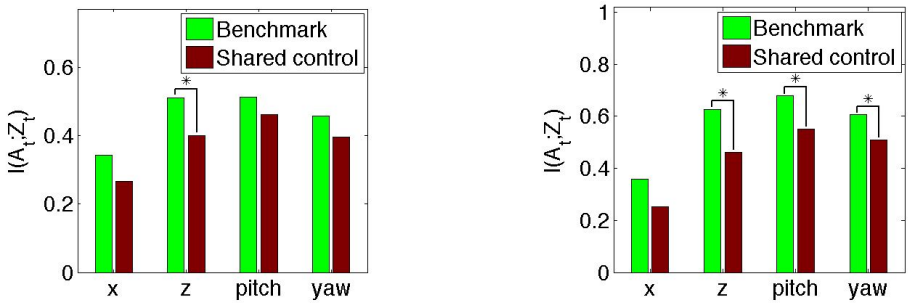
(a) *Participant 3.* ***(b)*** *Participant 5.* ***(c)*** *Participant 6.*

Fig. 4.21: *Visualizations of the learned proximity sensor usage for the final link. Transparency of square pyramid representing the field of view of a given sensor is made to vary with the mean Neural Network (NN) weights for sensor. High transparency indicates low usage, and vice versa.*



(a) *Average MT over participants.* **(b)** *MT for each participant.*

Fig. 4.22: *The Mean Time (MT) with and without (benchmark condition) the shared control.*



(a) *6 first seconds of trajectories.* **(b)** *6 last seconds of trajectories.*

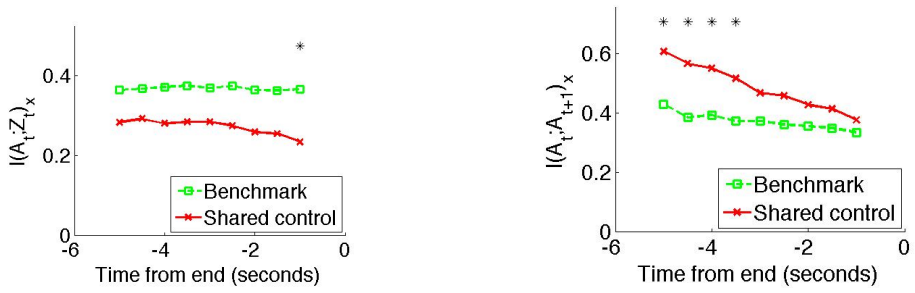
Fig. 4.23: *Measure for controllability, the mutual information over the velocity components for the noise added (Z) and the robot actuation (A). Calculated over 6 second periods indicated.*

4.4.4.3 Controllability from the User's Perspective

The results for the controllability from the user's perspective can be seen in Fig. 4.23. There was a decrease in the mutual information, and thus increase in the controllability, for all velocity components with the shared control. For the first 6 seconds of the trajectories, the difference for z was significant, with $t(5) = 3.34$, $p = 0.021$. For the last 6 seconds of the trajectories, the differences for z ($t(5) = 4.17$, $p = 0.009$), *pitch* ($t(5) = 2.65$, $p = 0.046$) and *yaw* ($t(5) = 2.98$, $p = 0.031$) were significant. Fig. 4.24a shows the same metric calculated over 2 second time-windows at different times before the end of the trajectories. The mutual information over the noise and robot actuation is lowered with the shared control at all times. Only the x component is shown, however similar reductions were seen in the other components.

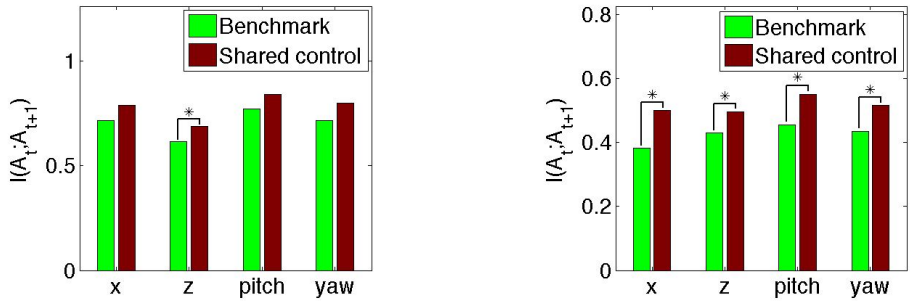
4.4.4.4 Predictability of Execution

The results for the predictability of execution can be seen in Fig. 4.25. For the first 6 seconds of the trajectories, the difference for z was significant, with $t(5) = -3.34$, $p = 0.021$. There was a statistically significant increase for all velocity components with the shared control for the last 6 seconds of the trajectories. A paired t-test was used, with $t(5) = -3.95$, $p = 0.011$ (x), $t(5) = -3.29$, $p = 0.022$ (z), $t(5) = -4.63$, $p = 0.006$ (*pitch*), and $t(5) = -2.86$, $p = 0.036$ (*yaw*). Fig. 4.24b shows the same metric calculated over 2 second time-windows at different times before the end of the trajectories. The predictive information is increased with the shared control, but the



(a) *Measure for controllability.* **(b)** *Predictability of execution.*

Fig. 4.24: *Information metrics calculated over 2 second time-windows centered at respective times before end of trajectories (x component only). Stars indicate statistically significant differences (at the .05 level).*



(a) *6 first seconds of trajectories.* **(b)** *6 last seconds of trajectories.*

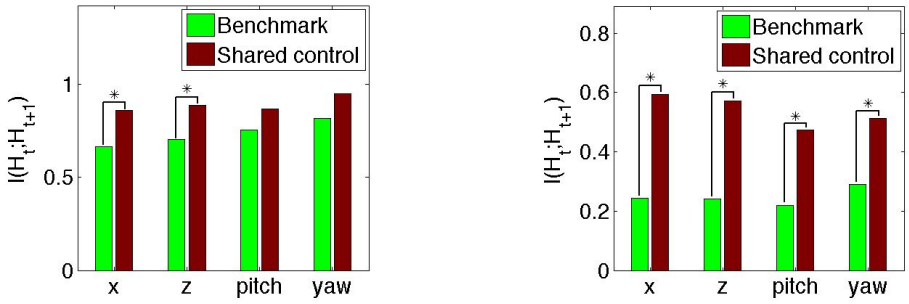
Fig. 4.25: *Predictability of execution, the mutual information over velocity components for the robot actuation (A) at two points in time. Calculated over 6 second periods indicated.*

effect is reduced towards the end of the trajectories. Only the x component is shown, however similar increases were seen in the other components.

As can be seen from Fig. 4.26, the able-bodied user input H also had an increase in predictability with the shared control, up to 144.4% for the x velocity component (Fig. 4.26b) over the last 6 seconds. A paired t-test was used, with $t(5)$ = -5.57, p = 0.003 (x), $t(5)$ = -5.48, p = 0.003 (z), $t(5)$ = -4.06, p = 0.010 ($pitch$), and $t(5)$ = -4.05, p = 0.010 (yaw). For the first 6 seconds the effect was smaller, but significant in the x ($t(5)$ = -2.82, p = 0.037) and z ($t(5)$ = -7.34, p = 0.001) components.

4.4.5 Discussion

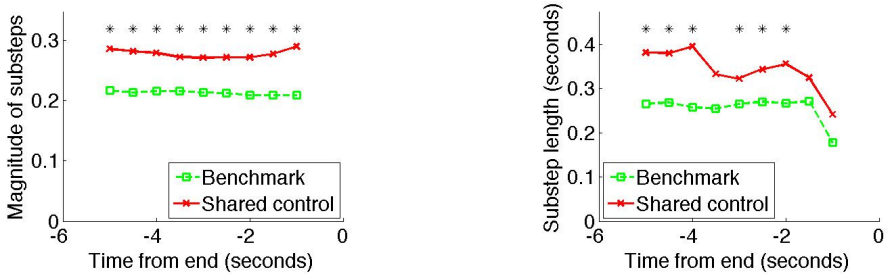
In general the results obtained show a strong consistency. That is, all three main metrics defined showed improvements in performance, as was also seen in the experiment with the simplified robot embodiment. This indicates that the approach followed for modeling and benchmarking has captured some important features of the impact of the shared control. The consistent improvement also gives an indication of the potential of the adaptive collision-limitation behavior, if we assume that the issues related to training and sleep are of minor importance to the results. Like with most task-oriented approaches, the learned proximity usage will not necessarily generalize to all task beyond what has been used for learning. It should have a positive effect on similar ones, though this should be explored further in the future.



(a) 6 first seconds of trajectories. *(b) 6 last seconds of trajectories.*

Fig. 4.26: *The mutual information over velocity components for the user input (H) at two points in time. Calculated over 6 second periods indicated.*

This increase in predictability is of particular interest, and was attempted explored further. As can be seen in Fig. 4.19, the user input typically consist of series of "jumps" or sub steps. If the approach has improved the speed-accuracy trade-off, one might expect that the user now has a more "confident" way of controlling the robot. A sub step was defined as any period of activity with a magnitude of user input above 5% of maximum, and the average absolute magnitude and length of the sub steps were calculated. As can be seen in Fig. 4.27, the sub steps with the shared control had a significantly higher absolute magnitude (though less so for *pitch*), and were also longer. Increasing the magnitude of the sub steps can be seen as taking advantage of a larger part of the range of possible user inputs, typically increasing the entropy of H. That is, increasing the amount of information that can potentially be transmitted from one point in time to the next.



(a) Magnitude of sub steps. *(b) Length of sub steps.*

Fig. 4.27: *Average absolute magnitude and length of sub steps in x component of user input, calculated over 2 second time-windows centered at respective times before end of trajectories. Stars indicate statistically significant differences (at the .05 level).*

4.5 General Discussion

The adaptive collision-limitation behaviour presented in this chapter has been presented at several international conferences, in journals, and in meetings with other researchers and industry representatives. From this interaction several questions and comments have been made about the reasoning behind the approach taken and its applicability to real-world assistive robots. In this section we will list and try to adress these critiques, hopefully also shedding some light on the ideas behind the work.

1. *Why Hebbian learning?* Even "simple" natural systems like ants show the ability to robustly operate in the real world, as individuals and as a collective, and to learn based on previous experience. For example combining path integration and landmark learning (Graham et al. 2010), or collectively adapting selection of nest based on previous experience (Sasaki & Pratt 2013). Modern robots can be seen as more "complex" than an ant, especially in terms of the computational power of the onboard computer, but struggle to perform and learn outside laboratory conditions. Are there concepts exploited by natural systems that can help simplify the learning process? A tight sensory-motor coupling is likely one, see quote from (Pfeifer & Bongard 2006) below:

   "If the robot is equipped with Hebbian learning, it does not want to associate; it is just doing so. Picking up correlations turns out to be especially useful because through sensory-motor-coordinated behavior, correlations are induced, as summarized in the principle of sensory-motor coordination (see, for example, Lungarella et al., 2005, which provides a quantative analysis building on the foundational ideas of Tononi et al. [1994, 1996]). In other words, these correlations most likely indicate underlying causal structure: e.g. the simultaneous sensory activity in both the force sensors in the muscles and the pressure sensors in the hand is caused by the act of grasping."

   A tight sensory-motor coupling thus enables actuation to be driven by the sensing, but also sensing to be driven by the actuation. The co-occurence of sensory stimulation with actuation can be used to drive the actuation in the future. For example to avoid certain forms of sensory stimulation, like pain. Hebbian associative learning is one way to represent this co-occurence, for example used in the DAC paradigm (Verschure et al. 1992), (Verschure et al. 2003), (Pfeifer & Bongard 2006). If such learning is sufficient to produce the desired change in behaviour, then any increments in algorithmic complexity should likely be justified based on the real improvements in performance of the behaviour. This will be part of the future work of this chapter.

2. *Assistive manipulators typically move slowly.* As mentioned earlier, the adaptive collision limitation behaviour was intended to aid the user in performing the speed-accuracy trade-off during robot control. If the robot moves much slower that what the user is capable of controlling on a given task, there there will necessarily be little need in helping the user to avoid collisions. Simply put, the user will have plenty of time to make sure collisions will not occur. Humans have been shown to perform targeted movements at relatively high hand velocity, for example 40-70 cm/s in (Sergio & Scott 1998). Most current assistive manipulators are much slower, with both AMOR and JACO moving at maximally 15-20 cm/s. However, most users of assistive manipulators are not operating at even close to the maximum robot velocity. See for example (Kim 2012), where standards pick-and-place tasks took on the order of 2 minutes for subjects with spinal chord injury that were regular users of powered wheelchairs. This is an order of magnitude slower than most non-disabled humans could perform such a task with their own hand. The author therefore believes the behaviour can be useful for current assistive manipulators when the throughput of the user control is low, or the task requires a high degree of accuracy. Like picking up objects from the back of a refrigerator shelf. The behaviour also seems highly applicable for future assistive manipulators that can safely be moved at higher velocities. That is, for soft and highly sensorized arms that mimic their biological counterparts.

3. *Is it inherently safe?* Any approach introducing autonomous actuation in an assistive manipulator will face the challenge of keeping the robot safe for the user. This is made more

difficult by the large range of different environments and tasks that can be faced in the user's daily usage of the robot. The collision-limitation behaviour described here is "passive", i.e. it only limits the user commands. This does not mean it can also cause undesired abrupt movements of the robot arm however. For example if an obstacle is suddenly lost from sight by the sparse proximity sensing. This can be alleviated by overlapping proximity sensing, for example using redundant modalities like infrared and ultrasonic sensors. As the user is always in the control loop, he/she can also adapt to such problems, perhaps even predict and avoid situations where they are likely to occur. This increases the safety of the approach, but making it "inherently safe" will require more investigation.

4.6 Conclusion

An adaptive collision-limitation behavior for assistive manipulators was presented. The approach has several interesting features with respect to previous work in shared control. It can be applied to high-DOF manipulator platforms operating in daily life environments. For example in situations where it is difficult to obtain accurate environment models and sensor to end-effector mappings. In addition, the adaptation is performed online using an unsupervised Hebbian learning. The learning rule associates a set of distributed proximity sensors with experienced collisions in real-time. The online nature of the adaptation seems important for maintaining the system predictable from the user's perspective. It is argued that the behaviour can help the robot operator improve the speed-accuracy trade-off on typical tasks, helping overcome the limitations of for example physical disabilities or time-delays (see Chapter 5) in the control loop. Two controlled experiments showed promising results. The experiments used able-bodied participants with simulated disabilities. Future work is needed to refine the experimental paradigm used, and to explore the application on a larger set of tasks. Including generalisation to unseen tasks and usage over a longer time-frame. Extensive testing with disabled users and a physical implementation of the system is ongoing and future work.

5. LEARNING TO LIMIT COLLISIONS DURING TELE-ASSISTANCE

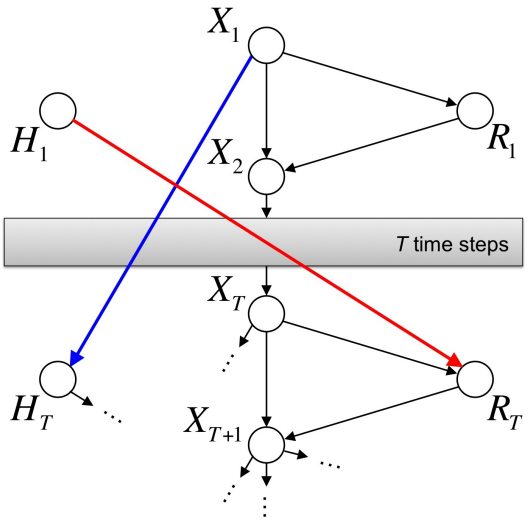


Fig. 5.1: *Directed acyclic graph representing the problem approached in this Chapter: A remote human operator H controlling a robot with actuation R over a (potentially varying) time-delay of T time-steps. Both operator commands and feedback are affected, and highlighted in red and blue, respectively. Subscripts indicate time-steps, see Section 2.2.3 for details.*

5.1 Tele-Assistance

The usefulness of assistive manipulators could be significantly increased by enabling a remote operator like a family member or a care provider to aid on the more complex tasks. This requires dealing with the time-variable delays of the internet, typically ranging from hundreds of milliseconds to seconds (Kamrani 2012), (Xiu et al. 2006). Two assumptions are common in work on teleoperation systems: 1) that exact models of the environment are available (from sensor data), and 2) that the required human movements can be explicitly modeled (Passenberg et al. 2010). Neither seem realistic for assistive manipulators, which may be used in any number of environments, such as in the user's home or in a grocery store, and on a great variety of tasks. However, we can assume that the operator will be performing targeted movements to objects in the environment. A controller that can help enforce the accuracy requirements of a teleoperation task can indirectly allow the operator to move with greater speed, and reduce the effect of the time-delays. Sampling the environment directly through distributed proximity sensing can be an effective way to estimate the robot's state, as in the previous Chapter. This section presents a

controller for teleoperating assistive manipulators based on these ideas, where the aid provided is gradually adapted to each operator's needs, extending the adaptive collision-limitation to tele-assistance. Thus attempting to cater for differences in the motor skills of different operators, in the time-delays for different user-operator pairs, and in the typical tasks on which the respective users require assistance.

5.2 Related Work in Teleoperation

One way to mitigate the effects of time-delays in teleoperation is to use predictive displays, which provide immediate feedback to the operator on the outcome of commanded actions through a virtual model of the robot. For example in space teleoperation (Lane et al. 2001), or in remote vehicle operation (Davis et al. 2010) with variable time-delays. A limitation of predictive displays is the requirement for accurate models of the robot and the environment in which it is used. Haptic force feedback allows human operators to perform complex tasks with physical contact in for example the medical field, in handling of toxic materials, and in outer space exploration (Basdogan & Srinivasan 2002). However, the variable time-delays, packet losses, and disconnections that occur over an internet connection can induce unstable forces, can degrade the performance, and can be harmful to the teleoperators (Xiu et al. 2006). Different time-delay compensation techniques have been developed to overcome these problems, such as wave-scattering theory (and the wave-variable approach) (Carignan & Krebs 2006) and Smith predictors (Rodriguez-Seda et al. 2009). Approaches that take into account information gained online about the remote Environment, the human Operator or the desired Task to be performed have been labelled EOT-adapted controllers (Passenberg et al. 2010). Such approaches are commonly used in advanced teleoperation systems to achieve an improved feeling of presence or to optimally increase task performance without comprimising stability. This is a similar concept to shared control. For example to reduce the velocity and force of impact (Everett & Dubey 1998). Haptic shared control has been shown to lead to performance improvements, but sometimes at the cost of the operator feeling that he/she is fighting the system (Abbink et al. 2012).

5.3 Haptic Aid on Short Time-Variable Delays

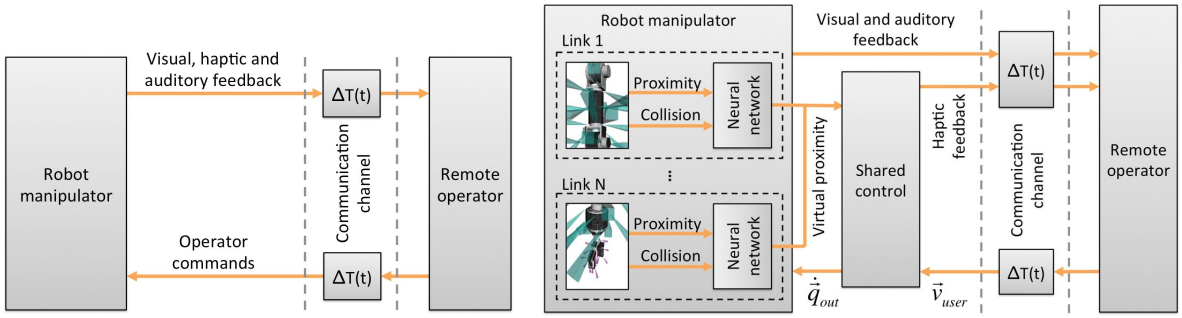
5.3.1 Introduction

This experiment was the first performed with the adaptive collision limitation behaviour in a tele-assistance scenario. A time delay of 300 ms was chosen, with a standard deviation of 30 ms. This is representative of typical internet time delays in a local environment, for example the round-trip time over 3G/4G mobile networks. The short time delay also helped the experimenters ease into the new paradigm, by ensuring that the length of the time-delay made the task easy enough to learn for the participants in a reasonable time (e.g. 1-2 hours).

5.3.2 Implementation

5.3.2.1 System architecture

The assumed system architecture can be seen in Fig. 5.2b. The remote operator has access to visual, force and audio feedback from the robotic manipulator, and uses this information to command the robot with Cartesian velocities $\vec{v}_{user}$. The control mode used here is further described in subsection 5.3.2.2. Both robot commands and operator feedback is passed over a communication channel with variable-time delay. The manipulator is covered with the same set of collision and proximity sensors as in the previous experiment (see section 4.4.2). The usage of these sensors in each link of the manipulator is regulated by a dedicated NN, which learns by

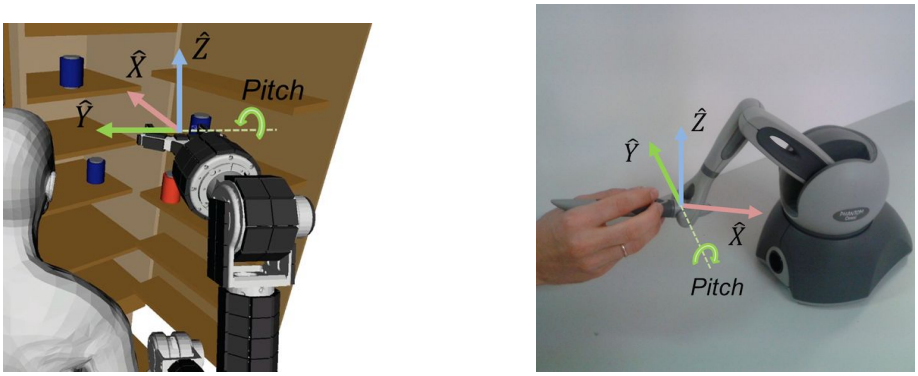


(a) Teleoperation with time-varying delay. *(b) The general architecture assumed here.*

Fig. 5.2: *An overview of the problem and the approach followed.*

associating proximity and collision sensors during collisions with the environment. The output of each NN is a set of virtual proximity readings, which are used in a shared control module. Here the robot is slowed down, and force/audio feedback is provided to the user.

5.3.2.2 Haptic Input Device Control Mode



(a) End-effector local frame. *(b) Stylus local frame.*

Fig. 5.3: *Visualization of the hybrid position/velocity control mode.*

Typical physical input devices for robotics can be broadly categorized as being based on velocities (e.g. joystick) or position (e.g. mouse). While position-based control can lead to higher performance, they have disadvantages such as less coordinated movements (Zhai 1998) and a limited workspace. A hybrid control mode based on both position and velocity input was here used, see Fig. 5.3. A Sensable PHANTOM Omni haptic device was assumed, which has 6 DOF position sensing and 3 DOF (x, y, and z) force feedback.

$$\vec{F}_{tot} = [F_x, F_y, F_z]^T = \vec{F}_{spring} + \vec{F}_{limitation}. \tag{5.1}$$

$$\begin{aligned} &\vec{F}_{spring} = k_{spring}\Delta\vec{P}, \\ &where: \\ &\Delta\vec{P} = [\Delta x, \Delta y, \Delta z]^T. \end{aligned} \tag{5.2}$$

The x, y and z displacements (with respect to a defined origen) of the stylus in the stylus-local frame was used to control the corresponding velocities of the robot end-effector. The *pitch*

angle of the robot end-effector matched the *pitch* angle of the stylus at all time. The total force provided ($\vec{F}_{tot}$ in Equation (5.1)) was composed of two components. First a spring-like component that always returned the stylus to the same position in space (the origen defined) if let go of, $\vec{F}_{spring}$, see Equation (5.2). Second, a component that provided feedback during assistance, $\vec{F}_{limitation}$. The latter is detailed in Equation (5.3). See Fig. 5.4 for a visualisation of the forces.

$$\begin{aligned} &\vec{F}_{limitation} = r_{norm}\vec{k}_{limitation}\Delta\hat{P}, \\ &where: \\ &\vec{k}_{limitation} = [k_x, k_y, k_z]^T. \end{aligned} \tag{5.3}$$

(a) *Spring-based force ($\vec{F}_{spring}$), before learning usage of proximity sensors.*

(b) *Spring-based force ($\vec{F}_{spring}$) and force based on virtual proximity ($\vec{F}_{prox}$).*

Fig. 5.4: *Visualization of the adaptive haptic feedback based on proximity.*

5.3.2.3 Resolving and Limiting Joint Velocities

Like with the experiment in the previous Chapter, the collision-limitation behaviour should work also for multi-link manipulators. There is therefore a need to propagate the commanded velocities to the sensors using the known kinematic structure of the robot. However, here the proximity ratio was used directly on the joint velocities resulting from the user commands. See Fig. 5.5. This meant that the inverse kinematics would only have to be resolved once, saving complexity and computational efforts. The effect is equivalent to limiting the Cartesian velocities, given that there is a fixed instantaneous mapping to the joint velocities, and that all the joint velocities are scaled by the same amount, r_{max}. That is, the Jacobian can be considered fixed for a given instantaneous robot pose.

5.3.3 Experiment Method

5.3.3.1 Participants

The participants were 9 undergraduate students of UC3M, 5 male and 4 female. A 10th participant was not able to finish all sessions due to other commitments, and was therefore not included in the analysis. All participants were right-handed. 2 had previous experience with 3D input devices. The mean age was 19.9, with a range from 19 to 21. Each participant was paid €10 for participation.

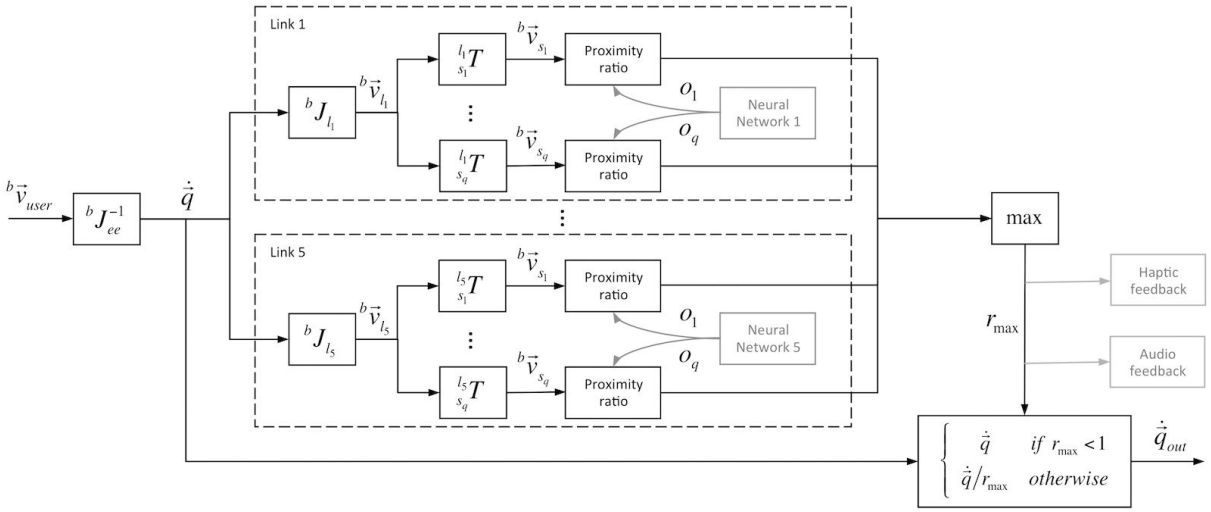
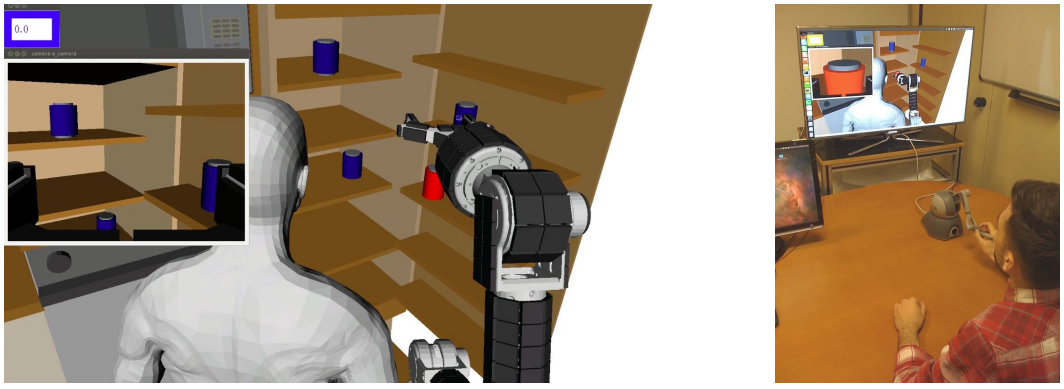


Fig. 5.5: *The approach used for resolving and limiting the multi-link robot joint velocity, as well as providing haptic (and audio) feedback. The b superscript denotes the robot base frame. Grey color indicates external modules.*



(a) *The view provided to the participants.* ***(b)*** *Physical setup.*

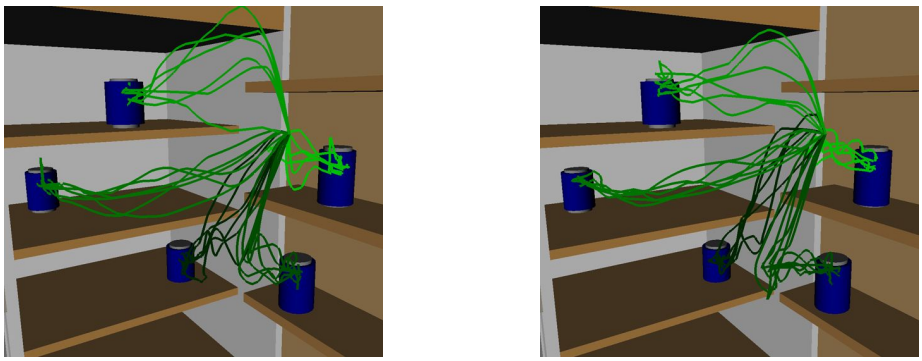
Fig. 5.6: *The simulated tele-assistance experiment setup.*

5.3.3.2 Simulated Environment and Time-Varying Delay

A tele-assistance scenario was simulated, as seen in Fig. 5.6. The OpenRAVE (Diankov 2010) virtual environment was used, running at approx. 50 Hz. The participants were given the simulated view from one camera mounted behind and above the wheelchair-user and one mounted on the end-effector of the robot. A time-varying round-trip time delay was simulated, with a mean of 300 ms and a standard deviation of 30 ms. The variation of the time delay was random, using a Gaussian noise low-pass filtered at 0.1 Hz. Given that the robot was virtual and there were no hard limits as to when the robot should react, the full time delay was added to the user input only.

5.3.3.3 Tasks Performed

The tasks performed involved moving the end-effector of the robot from an initial resting position (see Fig. 5.6a) to a pre-grasp position around one of 5 simulated cans in the virtual environment. For a given trial the target can was red, while the remaining were blue. A trial was



(a) *Benchmark.* ***(b)*** *Shared control.*

Fig. 5.7: *Example translational trajectories, for participant 6.*

automatically judged as completed when the two fingers of the robot end-effector was positioned around the thickest part of the can, stopped or with a small remaining velocity magnitude. The participants controlled the Cartesian x, z, *pitch* and *yaw* velocities of the robot end-effector, in the end-effector local frame. The timer changed color to red and incremented 10 seconds if any part of the robot collided with the environment, the physical model of the user, or any of the target cans (0.5 seconds minimum between collisions). For all trials the participants were instructed to attempt to achieve the lowest mean times possible, while keeping in mind that collisions were costly in terms of time.

5.3.3.4 Physical Setup

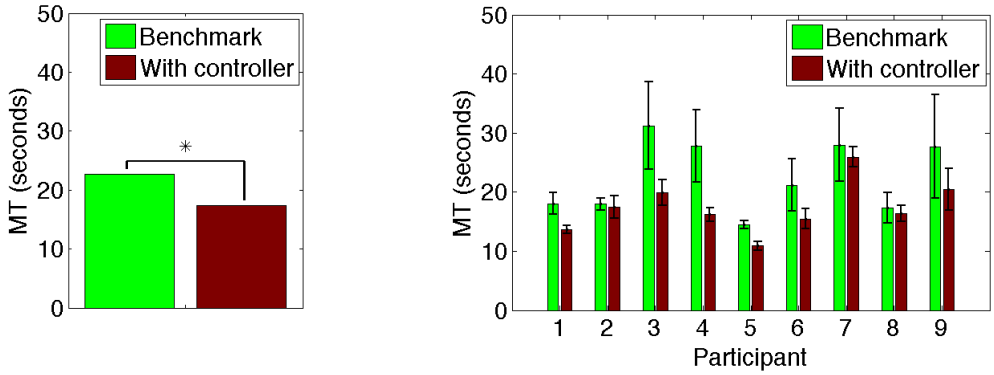
The physical experiment setup can be seen in Fig. 5.6b. The input device used was a Sensable PHANTOM Omni haptic device, with 6 DOF position sensing and 3 DOF (x, y, and z) force feedback. The two camera views simulated seen in Fig. 5.6a were displayed on a 40 inch (approx. 102 cm) display (Samsung 3D TV, UE40D8000), at a distance of about 2 meters. A colored timer was also shown.

5.3.3.5 Procedure

Each participant performed 3 days of testing, with about one hour of commitment each day. Each day consisted of 4 sessions. Each session had 15 attempts in total. The first day was used for training only. The second and third day the tasks were performed with our without the adaptive proximity-based haptic assistance. The order of the conditions were assigned randomly to each participant (balanced within-subject design). 2 training sessions were given before measuring performance for each condition, with performance being measured over the last 2 sessions only. In the assisted condition, the adaptation was only active during the training sessions. That is, each participant was told to attempt to achieve a comfortable level of assistance, and could decide when the training should be ended. Then the adaptation was disabled, and each participant was given 2 sessions to establish the performance using the NN weights learned during training (static).

5.3.4 Results and Discussion

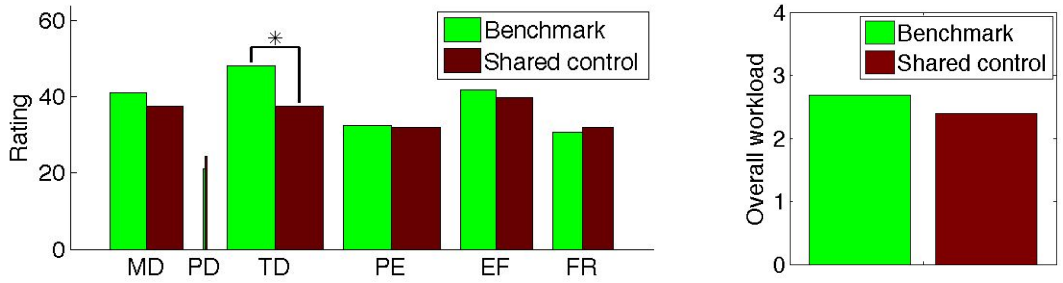
Fig. 5.7 shows the translations of the end-effector performed by one participant for the two conditions. A similar strategy seems to be used for solving the tasks with and without the proximity-based haptic assistance. The comparison of MT with and without the assistance can



(a) *Average MT over all participants.*

(b) *Individual MT for each participant. Error bars indicate standard error.*

Fig. 5.8: *The Mean Time (MT) with and without (benchmark condition) the aid of the controller. Based on the non-training sessions only.*



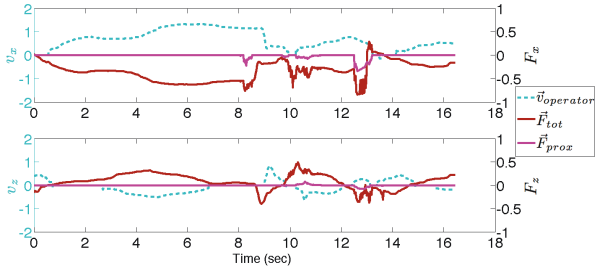
(a) *Ratings (height) and weights (width of columns). Lower rating is better.*

(b) *Overall workload. Lower is better.*

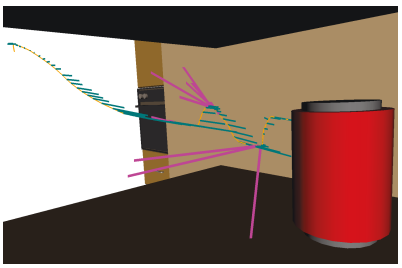
Fig. 5.9: *NASA-TLX subjective workload results.* MD - *Mental Demand,* PD - *Physical Demand,* TD - *Temporal Demand,* PE - *Performance,* EF - *Effort,* FR - *Frustration.*

be seen in Fig. 5.8. Here MT is the mean time measured over all attempts on all tasks by one participant. There was a statistically significant improvement in average MT over participants of 23% with the assistance. A paired t-test was used, with $t(8) = 3.842$, $p = 0.005$. As can be seen in Fig. 5.8b the reduction in MT varied from 2.8% to 41.7%, with no participants showing worse performance with the assistance. This is important, given that adaptation in a human-machine interface can also have a negative effect on the user's performance.

The results for the subjective workload measures (NASA-TLX) can be seen in Fig. 5.9. There was a significant reduction of 21.8% in the temporal demand (*TD*). A paired t-test was used, with $t(8) = 2.529$, $p = 0.035$. The 10.8% reduction in overall workload was not significant at the .05 level. Overall, the results indicate that the intervention of the system improved the quantitative performance, while also improving (or at least not significantly worsening) the qualitative experience from the user's perspective. This indicates that the approach holds some promise for mitigating this type of time-varying delay. Fig. 5.10 shows examples of the forces felt by the participants with the learned assistance, here for the final part of the trajectories. The point used as reference is the tool-point in between the fingers of the hand.



(a) Profiles of the x and z components of the velocity commanded by the operator, $\vec{v}_{operator}$, as well as of the total haptic force felt by the participant $\vec{F}_{tot}$ and the force component due to the aid by the controller, $\vec{F}_{prox}$.



(b) 3D trajectory, dark green lines indicate the $\hat{X}$ axis of the end-effector at each point (size indicates velocity in x), pink lines the aid-related haptic force, $\vec{F}_{prox}$.

Fig. 5.10: *One attempt by participant 6 on task 3 with the proposed controller: a) velocity and force profiles in x and z, and b) the end-effector trajectory.*

5.4 Exploring Longer Time Delays and Generalization

5.4.1 Introduction

A second experiment was performed, to test performance on a longer and more variable time delay, and to attempt some generalisation to unseen tasks. The time delay chosen, 600 ms, is close to that used in the literature, for example 80 and 480 ms mean one-way delays for bilateral haptic teleoperation over the internet (Rodriguez-Seda et al. 2009). Or for remote operation of a vehicle (Davis et al. 2010), where a mean variable round-trip delay of 700 ms was used (ranging from 200 to 1100 ms).

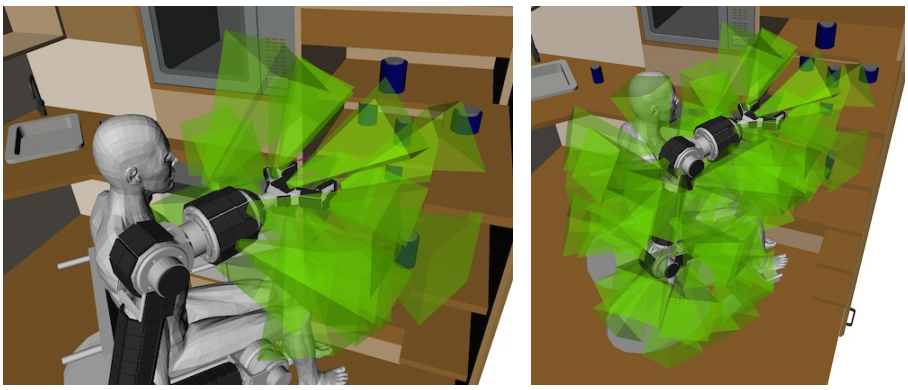
5.4.2 Implementation

There were three changes to the implementation from the first experiment. The first change made was to differentiate the collision and proximity activation in the audio feedback. That is, the collisions were signalled by low frequency pulses of static length. The help caused by the proximity sensors were still signalled by variable length pulses of higher frequency. The second change made from the first experiment was to limit the connectivity of the collision and proximity sensors. Sensors further away than 0.25 m and with a rotation that differed by more than 70 degrees were not connected. This to help avoid associating proximity sensors with collisions that occurred in the other extremes of a link. The third change was to replace all the proximity sensors with digital Silicon Labs Si1143 sensors. A great advantage of these sensors is that they work from approx. 3-5 cm to 40 cm. See Fig. 5.11 for the distribution used, and section 7.4 for the first physical implementation using these sensors.

5.4.3 Experiment Method

5.4.3.1 Participants

The participants were 8 undergraduate students of UC3M, 4 male and 4 female. 5 participants were right-handed, 3 were left-handed. 3 had previous experience with 3D input devices. The mean age was 23.0, with a range from 20 to 31. Each participant was paid €10 for participation.



(a) *Sensors on last link (wrist and hand).* ***(b)*** *All sensors.*

Fig. 5.11: *Field of view of simulated proximity sensors (Silicon Labs Si1143).*

5.4.3.2 Simulated Environment and Time-Varying Delay

A very similar teleassistance scenario as used in the first experiment was simulated, as seen in Fig. 5.12. See section 5.3.3.2 for details. A longer time delay was used, with a mean of 600 ms. It was also made much more variable, with a standard deviation of 120 ms. This meant the delay could vary from close to zero up to over a second. The variation of the time delay was random, using a Gaussian noise low-pass filtered at 0.1 Hz, as in the first experiment.

5.4.3.3 Tasks Performed

The tasks included the 5 tasks performed in the first experiment, but also 5 more target cans. The additional 5 cans were only used when the learning for the proximity-based haptic assistance was not active. Again, the tasks performed involved moving the end-effector of the robot from an initial resting position (see Fig. 5.13) to a pre-grasp position around one of the simulated cans in the virtual environment. See Section 5.3.3.3 for further details.

5.4.3.4 Physical Setup

The same physical setup as for the first experiment was used. See Fig. 5.12 and Section 5.3.3.4 for details.

5.4.3.5 Procedure

Each participant performed 3 days of testing, with about one hour of commitment each day. Each day consisted of 3 sessions. Each session had 20 attempts in total. The first day was used for training only. The second and third day the tasks were performed with our without the adaptive proximity-based haptic assistance. The order of the conditions were assigned randomly to each participant (balanced within-subject design). One training session was given before measuring performance for each condition, with performance being measured over the last 2 sessions only. In the assisted condition, the adaptation was only active during the training session. This condition was performed on the 5 original target cans. Each participant was told to attempt to achieve a comfortable level of assistance, and could decide when the training should be ended. Then the adaptation was disabled, and each participant was given 2 sessions to establish the performance using the NN weights learned during training, on all 10 cans.

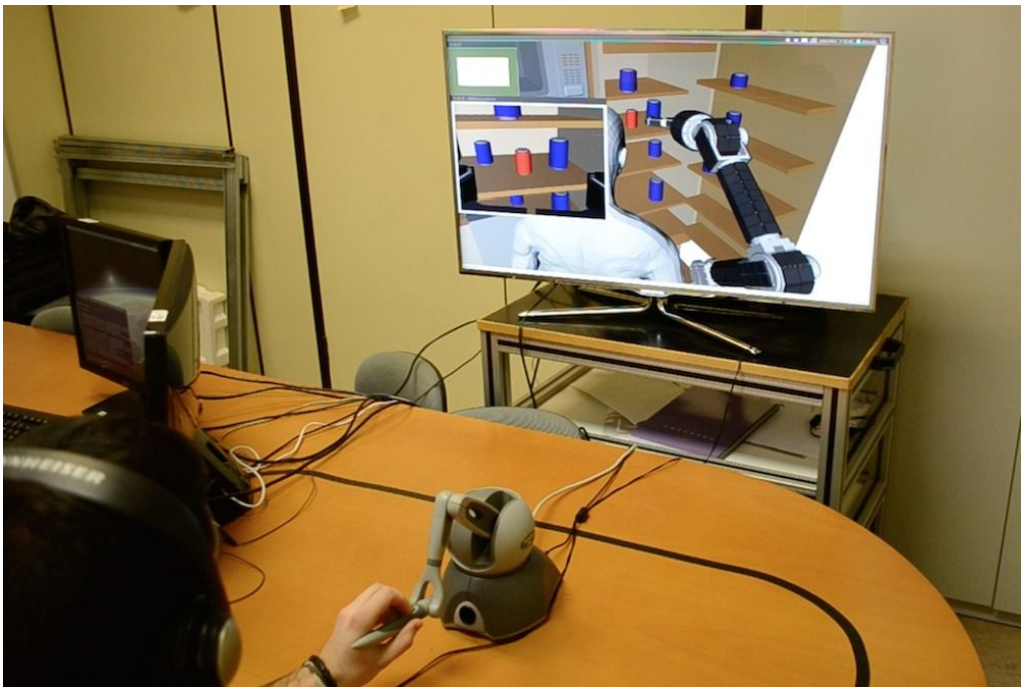


Fig. 5.12: *The setup for the second simulated tele-assistance experiment.*

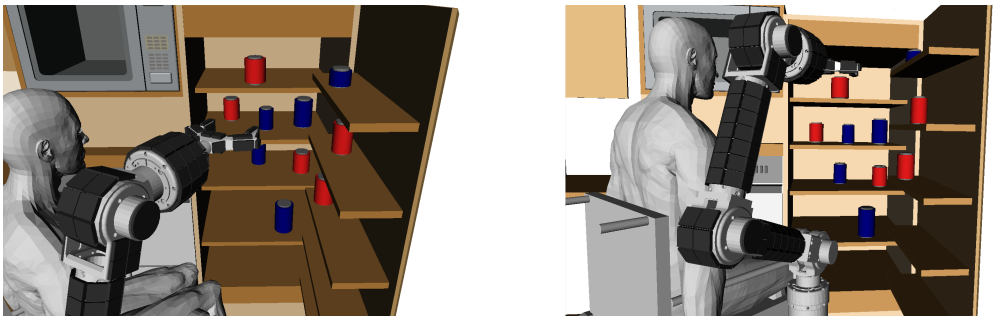
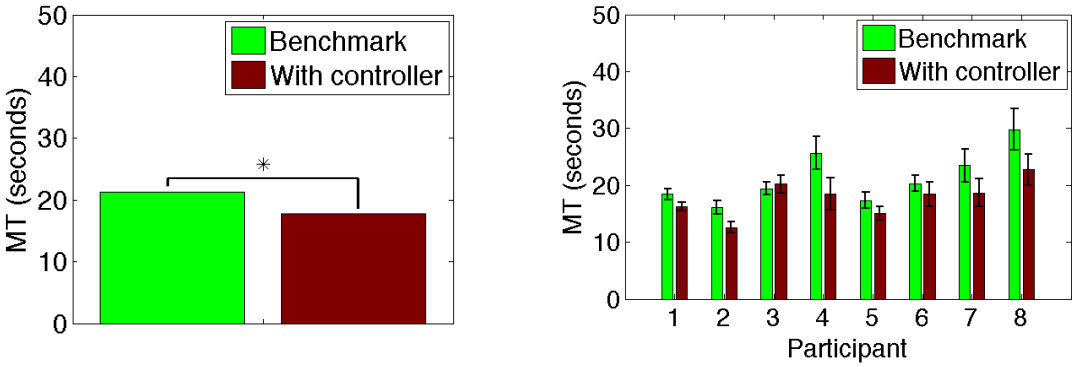


Fig. 5.13: *Two views of the tasks for the second experiment. Colors for visualisation only. The 5 red cans were used for both learning and testing, the 5 blue cans only for testing.*

5.4.4 Results and Discussion

The comparison of MT for the second experiment can be seen in Fig. 5.14. Here MT is the mean time measured over all attempts on all tasks by one participant. There was a statistically significant improvement in average MT over participants of 16.4% with the assistance. A paired t-test was used, with $t(7) = 3.633$, $p = 0.008$. As can be seen in Fig. 5.14b, one participant had a 4.1% increase in MT, while the remaining had a reduction (maximum reduction of 27.7% for participant 4).

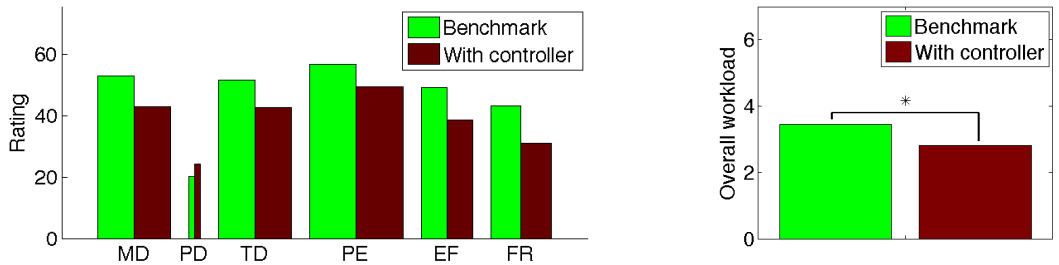
The results for the subjective workload measures (NASA-TLX) can be seen in Fig. 5.15. There was a significant reduction of 18.8% in the overall workload. A paired t-test was used, with $t(7) = 2.394$, $p = 0.048$. There were also (non-significant) reductions in all workload measures except the physical demand, the latter likely stemming from the increased resistance to movement with the haptic feedback.



(a) Average MT over all participants.

(b) Individual MT for each participant. Error bars indicate standard error.

Fig. 5.14: *The Mean Time (MT) with and without (benchmark condition) the aid of the controller. Based on the non-training sessions only.*



(a) Ratings (height) and weights (width of columns). Lower rating is better.

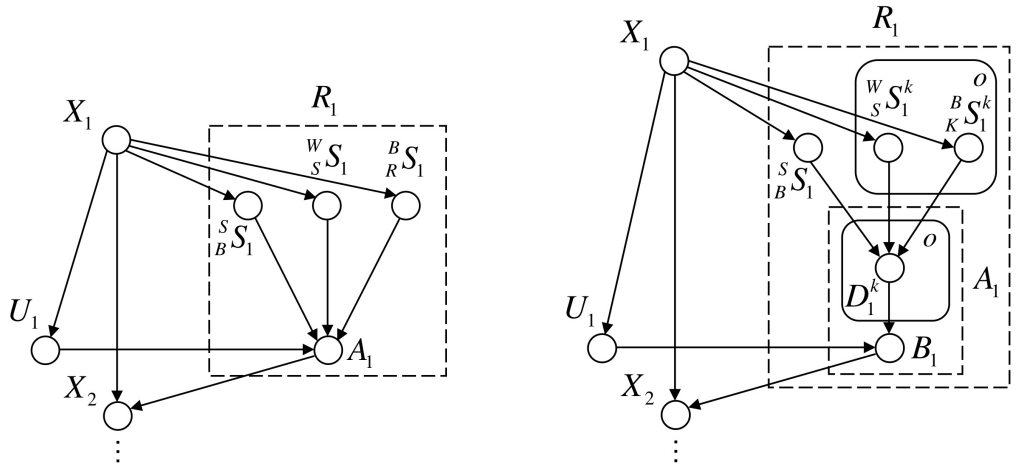
(b) Overall workload. Lower is better.

Fig. 5.15: *NASA-TLX subjective workload results.* MD - *Mental Demand*, PD - *Physical Demand*, TD - *Temporal Demand*, PE - *Performance*, EF - *Effort*, FR - *Frustration*.

5.5 General Discussion

So what are the potential benefits of an distributed approach in general, and in particular when attempting shared control on an assistive manipulator? As discussed briefly in the previous Chapter, enabling simple local learning approaches is likely one. Here we will discuss two more potential benefits, the reduced need for central information processing and internal representations.

1. *Less Central Information Processing.* To explore this issue further we will go back to the system models presented in Chapter 2 based on directed acyclic graphs and Information Theory. Fig. 5.16a is a more detailed representation of a centralised approach to sensing. The robot has been expanded to include three random variables representing sensor information, all of which influence the robot actuation A. The random variable ${}^{W}_{S}S$ represents the information obtained about the world with respect to a given sensor, for example the color and depth images from a 3D sensor like the Microsoft Kinect. Note that we here use pre sub- and super-scripts to denote the components that are being related by the variable: (S)ensor, (W)orld, (B)ase of robot, and (R)obot body. The random variable ${}^{S}_{B}S$ represents the mapping of the sensor with respect to the base of the robot, from a fixed location in the environment or from a mounting point on the wheelchair of the user. The random variable ${}^{B}_{R}S$ represents the mapping of the extremities of the robot with respect to the base, i.e. the joint encoder information and the robot kinematic model.



(a) *Generic view.* ***(b)*** *Expanded view.*

Fig. 5.16: *Directed acyclic graphs as system models of a hypothetical centralised approach to shared control of an assistive manipulator.*

We will here define the central information processing as that occurring when determining the overall robot movements in A to aid the operator in limiting collisions. For the centralised approach assumed above, there will be 4 channels of visual information, for example red, green, blue and depth. We will assume a 30 Hz update rate, a 640 by 480 image resolution and 8 bit pixels. See Fig. 5.17 for examples. This means 27.6 MB/s of data is received from ${}^{W}_{S}S$. For the random variable ${}^{S}_{B}S$ we will use the best case scenario, that the sensor is fixed with respect to the base of the robot and that the mapping is known. For ${}^{B}_{R}S$ the main source of information are the encoders. Assuming 7 encoders operating at 100 Hz and with a resolution of 1024 gives approximately 875 B/s. In general we can therefore assume that the central information processing will have to deal with on

the order of tens of MB/s.

For a distributed approach, the total information received by the collision and proximity sensors is already much smaller, on the scale of 50 kB/s if assuming 500 sensors in total, 8 bit depth resolution and 100 Hz update rate. However, in a physical implementation of the collision limitation behaviour the processing of the sensor data (and the learning) can be done in microcontrollers in each link. Thus the central processing may only be receiving the maximum proximity ratio from each link, which amounts to 700 B/s for a 7 DOF arm, plus the 875 B/s of data from the encoders. That is, under 1.6 kB/s in total, four orders of magnitude below that of the centralised approach.

The desire for keeping central information processing low may seem superfluous with the rapid development of processing power. However, a similar system could be envisioned in robots with much more limited power, mass and size requirements, such as human prostheses. In addition, any information received will have to be taken into account in the decision making process of the robot. That is, it makes the mapping from sensor input to actions more complex. Reducing it can likely help increase the robustness of the system.

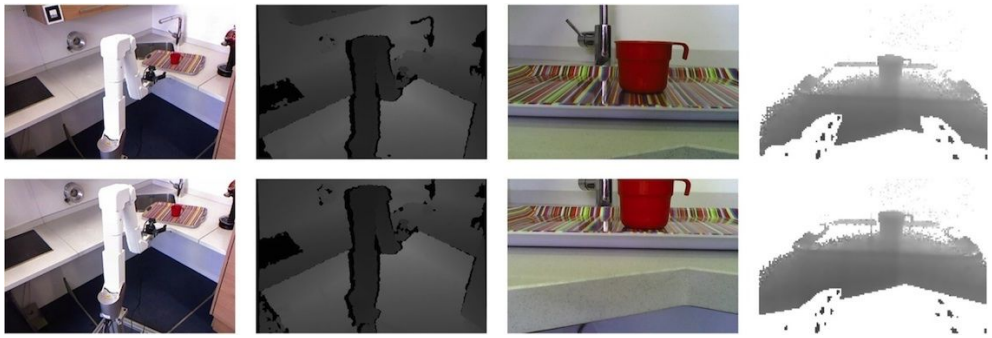
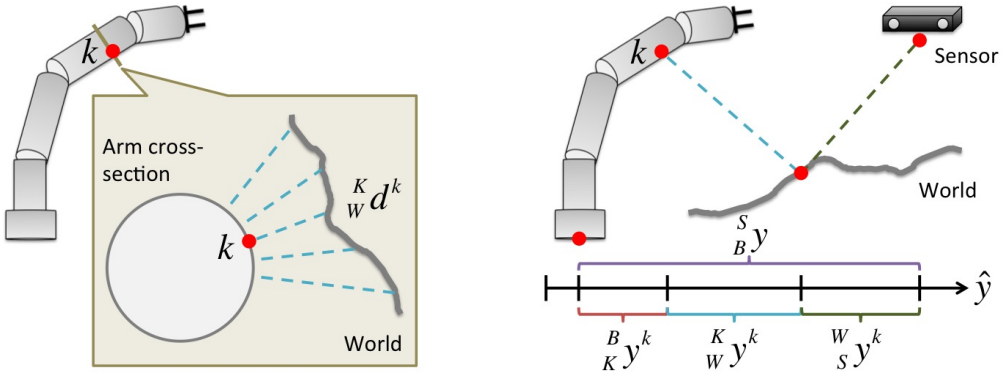


Fig. 5.17: *Color and depth sets, lower set rotated down by approx. 5 deg. Both static (Microsoft Kinect) and hand-mounted (webcam/PMD nano) sensors.*

2. *Simpler Internal Representations.* A robot will typically have to maintain an internal representation of itself with respect to the external world, on which it can plan its actions. However, for such a model to be useful it needs to be exact, and to be kept up-to-date. It is therefore desirable to reduce the complexity of the internal representations needed. This is also closely related to the concept of the "state" of the robot. For a behaviour aimed at limiting collisions, the state should describe the relationship of the robot's physical structure to the physical environment in which it operates. A close to minimal description for an assistive manipulator is shown in Fig. 5.18a. Here the state consists of the o normal distances d from the surface of each point k on the robot to the closest obstacle.

   Fig. 5.18b is a graphical representation of the relationships between the robot base, a point k on the robot, the world, and the external sensors, along an arbitrary axis $\hat{y}$. The principal objective of the system is to respond to the distance from point k to the world, here distance ${}^{K}_{W}y^{k}$. Considering again Fig. 5.16a, we can consider that the uncertainty in the robot internal state is given by the joint entropy of the random variables representing sensory information. That is, by $H_{cent} = H({}^{W}_{S}S, {}^{S}_{B}S, {}^{B}_{R}S)$ for the centralised approach. If we consider the three random variables to be independent then we can write:

$$H_{cent} = H({}^{W}_{S}S) + H({}^{S}_{B}S) + H({}^{B}_{R}S). \tag{5.4}$$



(a) *A collision-limitation "state": normal distances to the world.*

(b) *Example $\hat{y}$ axis distances between components for point k.*

Fig. 5.18: *Example robot-world relationship for an assistive manipulator.*

Now going into more detail, the centralised approach can also be modelled as in Fig. 5.16b. Here the actuation A is expanded to include a random variable B that makes the decisions about what movement to perform based on the o calculated distances to the environment D. These distances depend on the general mapping of the external sensor to the robot base ${}^{S}_{B}S$, but also the point-specific distance from the base of the robot ${}^{B}_{K}S^{k}$, and the corresponding distance from the sensor to the point detected in the environment ${}^{W}_{S}S^{k}$. This is also in accordance with the axis-specific visual representation in Fig. 5.18b. For each point k the entropy along the $\hat{y}$ axis can then be written as in Eq. (5.5).

$$H(Y^{k})_{cent} = H({}^{W}_{S}Y^{k}) + H({}^{S}_{B}Y) + H({}^{B}_{K}Y^{k}). \tag{5.5}$$

It can be seen that the entropy of all points is influenced by the uncertainty in the sensor to robot base mapping, $H({}^{S}_{B}Y)$. Thus any alteration in this mapping, for example the sensor being moved, will influence the uncertainty of all the calculated measurements. See for example the 5 degree pitch rotation in Fig. 5.17. At 2 m distance there can be errors on the order of 17.5 cm. This is likely also at least partially true for the sensor data itself, i.e. a fingerprint smudge on the lens will affect many of the calculated distances from the robot to the environment.

We can now compare this expression for entropy with that for a distributed approach in Eq. (5.6). This is based on the system in Fig. 5.19, which is a more detailed representation of the distributed approach to sensing used here. The robot actuation A is influenced by l random variables L, one for each link h. Each L is driven by input from the n proximity ratios (each corresponding to one collision neuron/sensor), here denoted V. As described before, each proximity ratio is influenced by one collision sensor C, by up to m proximity sensors P, and by the pose of the collision sensor with respect to the base, represented by ${}^{B}_{C}S$.

$$H(Y^{k})_{dist} = H({}^{K}_{W}Y^{k}) + H({}^{B}_{K}Y^{k}). \tag{5.6}$$

As the sensors directly measure the distances to the environment, here assumed to be ${}^{K}_{W}y^{k}$, the principal internal representation required is the mapping from the robot base to each point (sensor). Any inaccuracy in this mapping for a given sensor will only affect the entropy of that specific sensor, unless it is a link- or manipulator-wide change. Note that

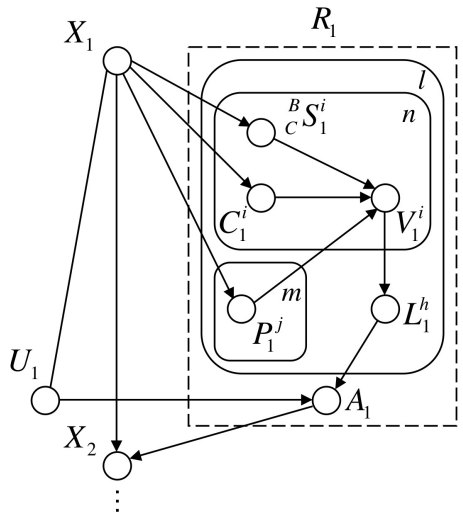


Fig. 5.19: *Example distributed decision making for the shared control proposed.*

we assume a sufficient coverage of proximity sensing to cover all o points required for a given task. In Fig. 5.17 is also shown the sensor data from in-hand mounted image sensors, for a visual comparison. At 0.5 m there can be errors on the order of 5 cm (tangential to the rotation) with a 5 degree rotation. However the overall distances to the objects from the robot are largely unchanged.

5.6 Conclusion

This chapter extended the adaptive collision-limitation behaviour to a tele-assistance scenario with time-variable delays in the control loop. The general approach will for the remainder of this thesis be denoted as Shared Distributed Adaptive Control (SDAC), given the inspiration from the DAC paradigm. Improvements were found both in the time to perform the tasks and in the subjective evaluation of the system, indicating that the results in the previous Chapter were not necessarily only caused by the simple stochastic disability simulated. There were also indications that SDAC can generalise to unseen tasks that are similar to those where the system was trained. Much broader studies are needed to be able to say exactly how much can be generalised. It may be that different sets of weights would have to be used in different settings, for example in different rooms of a user's home. However, this introduces another problem that requires investigation, how to choose or to merge different sets of weights. It would also be critical to study how this affects the predictability from the user's perspective. The improvements in subjective workload with SDAC are especially encouraging, as the acceptance of the user can often be the achilles heel for automation, and adaptation, in human-machine interaction.

6. LEARNING TO PREDICT AND IMITATE THE EXECUTION OF TASKS

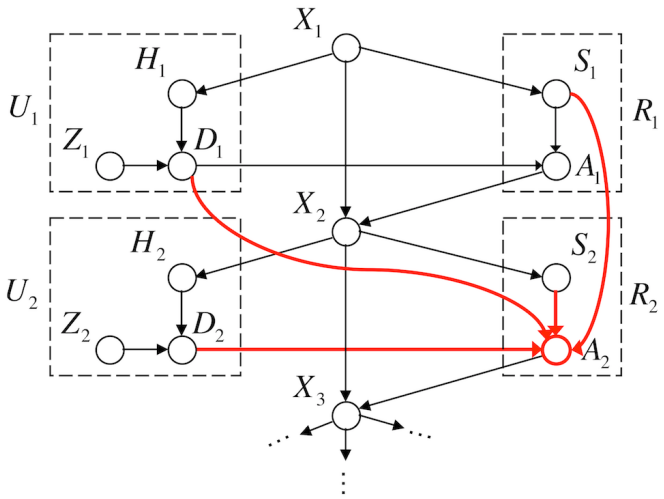


Fig. 6.1: *Directed acyclic graph representing the problem approached in this Chapter: Learning robot actuation A that takes into account high-dimensional sensory information S and noisy user input D from the current time, and the recent past. Subscripts indicate time-steps, see Section 2.2.3 for details.*

6.1 Introduction

The previous two Chapters focused on assisting the user in avoiding collisions. However, it might also be possible to help the user reach the goals he/she would like to reach. That is, for the robot to predict the intent of the user, and provide assistance on fulfilling this intent through actuation. This would likely require some form of imitation learning by the robot, as the system should adapt to a specific user and his/her way of performing tasks. However, each of the demonstrations may differ due to the variability of the motor system, even more so for a physically disabled user. The solution therefore needs to be able to extract the sensorimotor "laws" underlying the demonstrations, in effect averaging out the influence of the noise Z. If this is achieved then the result may be a reduction in both cognitive and physical effort, in a sense a system that completes the thought of the user. The present Chapter will deal mainly with learning, prediction and actuation, while Chapter 7 will attempt to use the predictions to assist the user, as part of a shared control scheme.

6.2 Related Work

Humans and animals are able to learn while interacting with the environment and with other agents, and from this interaction develop the ability to recognize and label complex tasks. However, there are several challenges that must be overcome when attempting to transfer these abilities to a robot. The first is that most current imitation learning approaches are based on a hierarchy of modules representing different motor primitives. Including approaches based on Dynamic Movement Primitives (DMP), see for example (Pastor & Righetti 2011). Or recent work on a hierarchical Reservoir Computing (RC) network for imitation learning (Waegeman et al. 2009). The use of individual modules for each primitive typically requires additional centralized functionalities for gate-selection. It also seems to suffer from the symbol grounding problem (Harnad 1990), the need for grounding the symbols used to represent thoughts and beliefs in something other than just more symbols. The grounding of language in actions and sensorimotor knowledge is for example widely studied (Glenberg 2007),(Marocco et al. 2010). There are alternative approaches however. For example Recurrent Neural Networks (RNN) with parametric biases (Tani 2003). Here sensorimotor trajectories and their mapping to a set of parametric biases were learned through an offline supervised learning. This enabled multiple behaviors to be learned by the same RNN, each of which which could be recalled with the parametric bias values obtained. Or more recent work on Multiple Time-scales Recurrent Neural Networks (MTRNN) (Yamashita & Tani 2008). Here it was demonstrated that a functional hierarchy can emerge, enabling the segmentation of continuous sequences into movement primitives without explicitly modeling each primitive.

A second challenge is the large increase in the size of the number of possible distinct system states with increasing dimensionality. Local approaches may help avoid this "curse of dimensionality". One popular approach to local learning is based on Locally Weighted Learning (LWL) (Schaal & Atkeson 1998). Online multi-map regression approaches also exist, for example applied to learning a Finite State Machine (FSM) of subtasks and their policies (Grollman & Jenkins 2010). However, there is typically still a need to explicitly model each movement or primitive. A third challenge is that animals and humans are able to adapt their behavior while interacting with the environment. This may also be a useful skill for robots in applications like assistive robotics. Online intent prediction has been explored in the context of mobile assistive robot platforms (Demeester et al. 2008). Other work focused on online recognition of manipulator actions, through a modified formulation of DMPs (Akgün et al. 2010). Here the 3D trajectories of a human actor were learned for multiple objects and multiple behaviors per object, and then used to recognize the action online. Each action was encoded using an explicit model and the learning was performed offline however. Incremental imitation learning approaches do exist, for example using a Gaussian Mixture Regression (GMR) to represent each task (Calinon & Billard 2007). Here the learning is performed after each demonstration, and on sensor data that has first been through a dimensionality reduction. Learning in parallel with a demonstration has been suggested as a potential application for Liquid State Machines (LSMs) with online linear regression (Burgsteiner 2006).

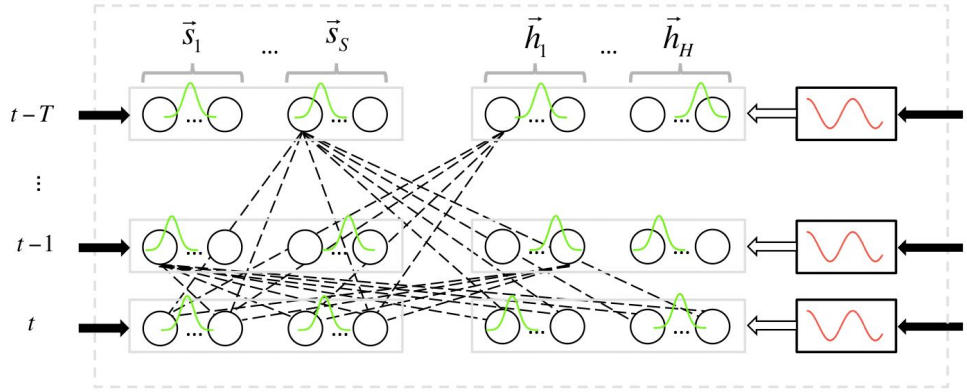
6.3 Hebbian Learning over Time-Delayed Inputs

A spatiotemporal connectionist NN was developed. Such networks are capable of handling patterns distributed across both space and time, see reviews in (Kremer 2001) and (Barreto et al. 2003). The goal of the work presented in this section was to explore learning in a high-dimensional robotic system, where both recognition and learning occurs online during demonstrations, and with an implicit grounding of the high-level representations of different movements.

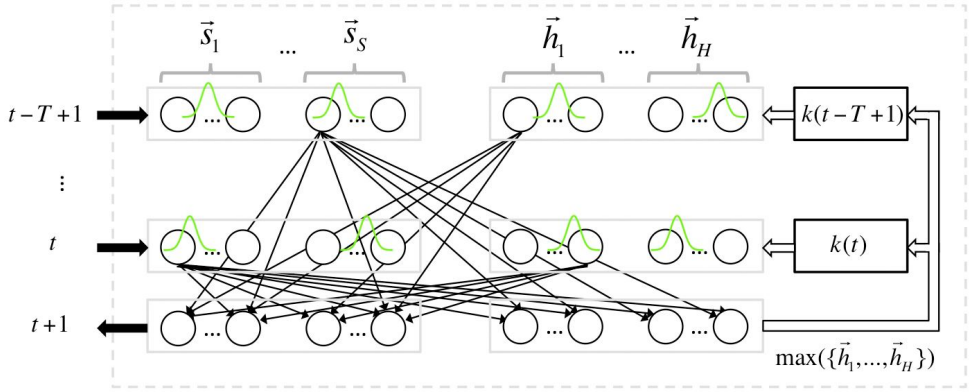
6.3.1 Neural Network Structure and Embedding

6.3.1.1 Overview

The NN consists of a number of input layers, each with a given input delay. These input delays range from 0 seconds (the present) to one or several different time-steps into the past. See Fig. 6.2. Each sensor is represented by a set of neurons, $\vec{s}$. Separate sets of neurons represent the different hypotheses the agent may have, each set written as $\vec{h}$. Each neuron is assigned a specific sensor/hypothesis value, and the set of neurons used can thus be seen as a discretization of the potentially continuous sensor and hypothesis inputs. The set of neurons representing the discretization of a sensor/hypothesis is activated according to a Gaussian curve, with a mean at the actual sensor/hypothesis value and a set standard deviation. All the sensors in layers in the past are fully connected to the other sensors in the layer representing the present (at time t). No connection exists between the same sensor in the past and present. The hypotheses in layers in the past are fully connected to all the sensors in the layer representing the present. No connection exists between the hypotheses in the past and in the present. The NN has two operating modes, learning and prediction, which are detailed next.



(a) *Learning mode. Dark dashed lines indicate Hebb-like learning.*



(b) *Prediction mode. Thin dark solid arrows indicate activation propagation.*

Fig. 6.2: *The Neural Network (NN) architecture used. For clarity of presentation only the synapses for one neuron in each past layer is shown. Grey solid boxes represent input layers, while the grey dashed box indicate the border of the NN. Thick solid arrows represent input from the outside while thick hollow arrows represent input originating inside the NN. Example Gaussian activation curve for each set of neurons shown in green.*

6.3.1.2 Learning Mode

In the learning mode there is input to all layers, from present time to T time-steps into the past, as shown in Fig. 6.2a. That is, the input layer most in the past receives inputs delayed by T. No activation is propagated, and a Hebb-like (Hebb 2005) learning rule is applied across all connections, shown in Eq. (6.1). The rate of change of the synapse weight, Δw, is proportional to the learning rate η, as well as the activation for the neuron in the past, c_i, and the neuron in the present, p_j. In addition, the rate of change of the same neurons, Δc_i and Δp_j, is taken into account. The mean of the two rates is used. This serves as a crude method for reducing over-learning when a sensor is static.

$$\Delta w = \eta c_i p_j \left(\frac{\Delta c_i + \Delta p_j}{2} \right). \tag{6.1}$$

6.3.1.3 Prediction Mode

In the prediction mode, the inputs are all shifted back in time one step. See Fig. 6.2b. That is, the present time layers do not have any input, while the layers most in the past now receive inputs delayed by $T - 1$. The input from all past layers are then propagated over the synapses, with the corresponding learned weights, to create activation in the bottom-most layers in the figure. This activation is then used as a prediction of the future activation at $t + 1$. A simple linear neuron transfer function was used. The value represented by the neuron with the maximum activation for a given sensor/hypothesis was used as the prediction for that sensor/hypothesis.

6.3.1.4 Input Types

The NN is designed to handle different types of inputs. From continuos inputs, for example sensors like robot joint encoders, to discrete inputs. For example higher-level concepts like wether an object is round or square. Uni-valued inputs can be represented by one neuron. The hypotheses fed to the system as inputs were each given a single numerical value. The representation of each hypothesis by the system was spread over a range of neurons however. A pattern generator was then used to add a time-variable pattern to the value given to each hypothesis. A sinusoidal pattern was selected for the implementation presented here, as seen in Fig. 6.2a. Central Pattern Generators (CPG) are known to be important in human and animal movements, for example for generating locomotion. It was not the purpose of the present study to model or attempt to emulate biological CPGs. However the tight coupling between sensory information and CPGs (Ijspeert 2008) is of interest, and serves as inspiration. There is also an ongoing discussion about the similarities between CPG circuits and some circuits in the neocortex (Yuste et al. 2005).

6.3.1.5 Hypothesis Feedback

If there is strong evidence (high activation) for a given hypothesis, this can likely be used to also improve the future prediction of the state. One way to achieve this is by the hypothesis feedback shown in Fig. 6.2b. Here the prediction of the hypothesis is fed back into the NN at the corresponding times in the future by means of delayed input lines. The value used to represent the current hypothesis is simply the value for the neuron with the maximum activation. The strength of the hypothesis, k, was also estimated, as seen in Eq. (6.2). The strength is zero if there is no difference between the two hypotheses with highest activation, max_{hyp1} and max_{hyp2}, and if there is no difference in the max and mean activation for the neurons representing the

hypothesis with the highest activation, $hyp1$. The activation caused by the hypothesis feedback was then made to vary proportionally to this measure. A zero strength caused zero activation.

$$k = \beta(max_{hyp1} - max_{hyp2})(max_{hyp1} - mean_{hyp1}). \tag{6.2}$$

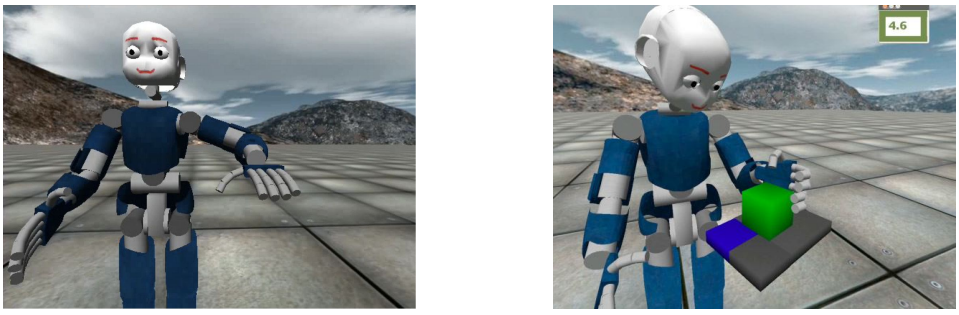
6.3.2 Trajectory-based Benchmarking

6.3.2.1 Introduction

The goal of this experiment was to benchmark the ability to ground different types of high-level symbols in sensorimotor trajectories online, and to then predict the future. Simple number 8 shape movements with no physical interaction was used to maintain a high degree of control, and replicability, in the trials. These movements will be referred to as figure-8s in the rest of the text.

6.3.2.2 Method

Robot Embodiment The trials were performed on a simulated iCub robot (Tikhanoff et al. 2008), (Tikhanoff et al. 2011). See Fig. 6.3a. The left arm of the robot was actuated using an iterative inverse kinematics solver based on the KDL library from the OROCOS project (Bruyninckx 2001). Only 6 DOF (of 7) were actuated to keep the joint trajectories consistent, with the shoulder roll joint locked at 30 degrees. The YARP (Metta et al. 2006) communication protocol was used to communicate with the iCub simulator.

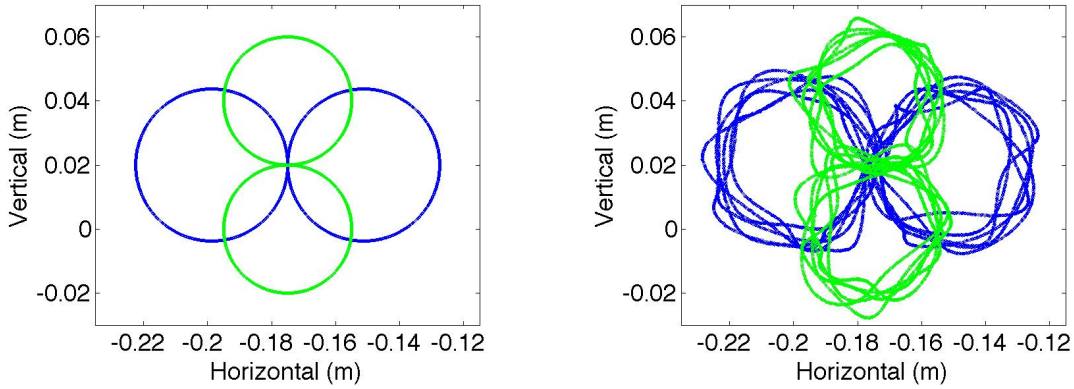


(a) *Benchmarking setup.* ***(b)*** *Object interaction setup.*

Fig. 6.3: *The simulated iCub robot for the two setups used.*

Tasks Used Two different movements were performed, both figure-8s, one vertical and one horizontal. See Fig. 6.4a. The respective radii of the circles used for each were 20 and 24 mm. The completion of each figure-8 took 15 seconds. 4 discrete labels were associated with the trajectories. The "Height" label was 1 in the top half of the trajectory, with respect to the vertical center of each figure-8, and 0 in the bottom half. Similarly, the "Side" label was 1 and 0, on the left and right half, respectively. The "HeightV" and "SideV" labels represented the direction of movement with respect to the same reference frame. Finally, the orientation of the figure-8 used, vertical or horizontal, was encoded as two "Task" hypothesis inputs: "Task 2" and "Task 1", respectively. For a set of trials noise was added to the trajectories used for learning. This was white noise with a Gaussian distribution, low-pass filtered at 0.5 Hz. See Fig. 6.4b.

Neural Network Setup The NN was setup with inputs at 4 different delays (1, 2, 3 and 4 seconds). A total of 446 neurons for the input at each time delay was used, with 60 neurons



(a) *Baseline.* ***(b)*** *With noise added.*

Fig. 6.4: *The figure-8 trajectories learned during benchmarking. Vertical (light green) and horizontal (blue) shown overlapping.*

per arm joint. The discrete labels were represented by 9 neurons each, and the two hypotheses by 25 neurons each. Limited joint ranges were used, see respective plots. Time-indexed buffers running at approx. 100 Hz were used to emulate the time-delays. A learning rate, η, of 0.0002 was used. The NN updated at approximately 8 Hz during learning and 45 Hz during prediction, on an 8-core Dell i7-2600 @ 3.4 GHz. For the hypotheses, 15 second period sinusoidal waves were used as patterns, matching exactly the period of the complete figure-8's.

Experiment Procedure The experiment was setup to perform both learning and prediction online. A set of Cartesian trajectories was created beforehand, with and without noise according to the description above. The relevant trajectory was sent to the inverse kinematics solver used to control the arm. The sensor readings from the arm encoders, as well as the labels/hypotheses given, were sent to the buffer used to delay the inputs. Each trajectory lasted 90 seconds, with the robot moving continuously. Learning was performed on the horizontal figure-8 or both, with or without noise. Prediction was then performed on horizontal figure-8s only, and included trials with horizontal figure-8s that were 25% faster or 25% smaller. Only arm joint encoder input was used to generate activation during prediction.

6.3.2.3 Results

An example of the weight matrices learned can be seen in Fig. 6.5. Each weight matrix represents all connections between the layers at one time in the past and the layers in the present. Each "blob" of activity roughly represents the relationship between one sensor/hypothesis in the past and one sensor/hypothesis in the present (over the respective neurons used for these sensors/hypotheses). The figure-8 shape can be distinguished in some blobs, although there is no Cartesian sensor information present.

A comparison of the actual and predicted trajectories for the baseline condition can be seen in Fig. 6.6. The predictions performed at 1 second in the past matches well the actual values, for both arm joint trajectories, labels and hypothesis patterns. No labels nor hypotheses were given during prediction, i.e. all predictions are performed on the basis only of the arm joint encoder input. It can be seen that the pattern used for the hypotheses had a period that exactly matched the period of the figure-8 (15 seconds). This was a conscious decision, but it is clear that it is also the best case scenario. A periodic movement task will at some point pass through the same

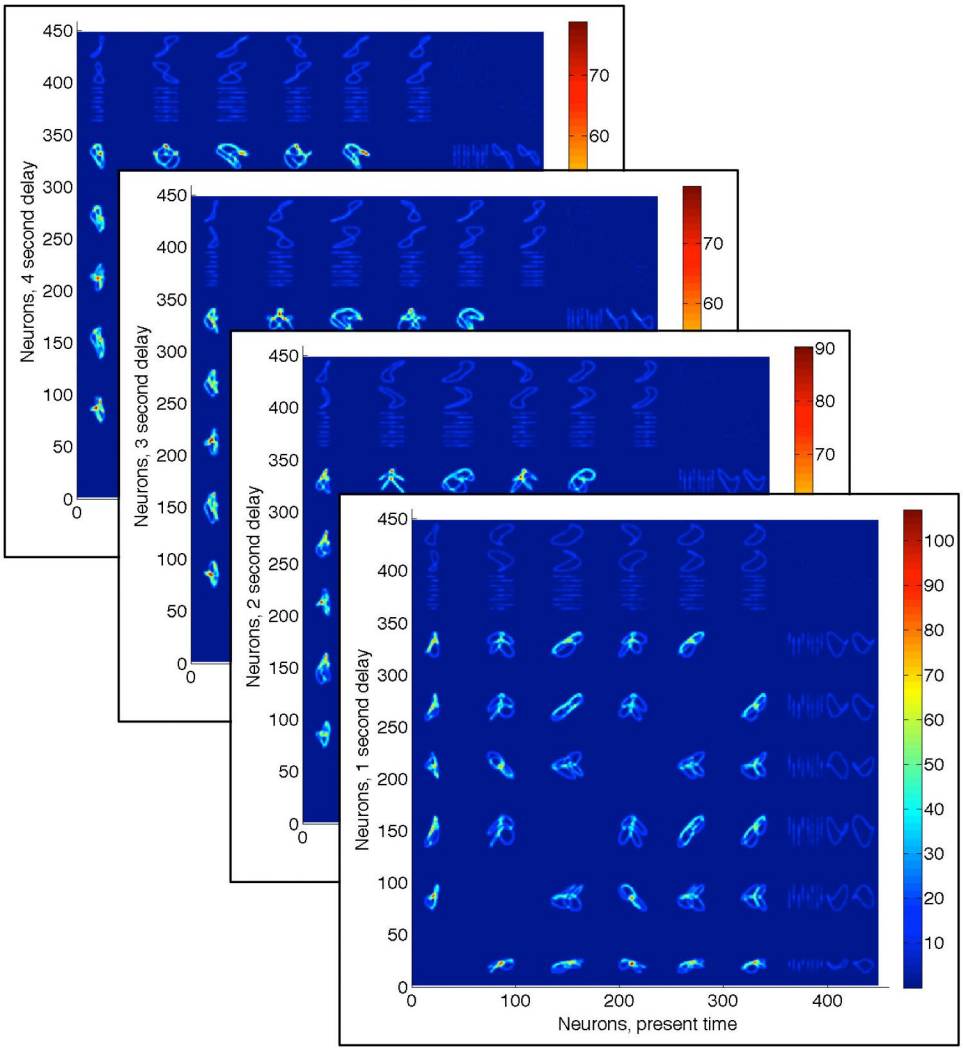


Fig. 6.5: *Learned inter-time weight matrices during benchmarking. After one horizontal and one vertical figure-8 trajectory of 90 seconds, with no noise added. 446 neurons on each axis. Neuron 0 - 360: Six arm joint angles. Neuron 360 - 396: Four discrete labels. Neuron 396 - 446: Two "Task" hypotheses.*

points in sensor space. It is here assumed that the agent has the ability to detect when this occurs and to provide a suitable period pattern for the hypotheses. With a non-synchronized period there will be ambiguity, growing with the number of periods used.

Finally, Fig. 6.7 shows the complete results for the benchmarking trials in average Root Mean Square Error (RMSE) over joint angles and the average percentage correctly predicted labels. The prediction of the joint angles were affected by both learning conditions and prediction conditions. The noise had a smaller effect than the use of one vs two different figure-8s. For the worst case scenario used the RMSE for the joint angles was kept within 6 degrees and the labels were predicted correctly more than 80% of the time. This indicates that the approach is reasonably robust to noise and to the ambiguity introduced by partially overlapping trajectories. The 25% speed increase had a quite consistent negative impact, while the effect of the 25% reduction in size was more ambiguous. There was an average reduction of RMSE of 15% when the hypothesis feedback was added, with the largest reductions for the more complex conditions.

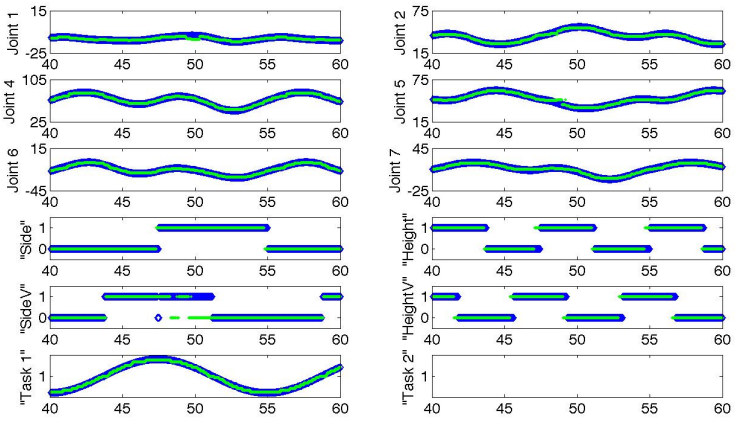
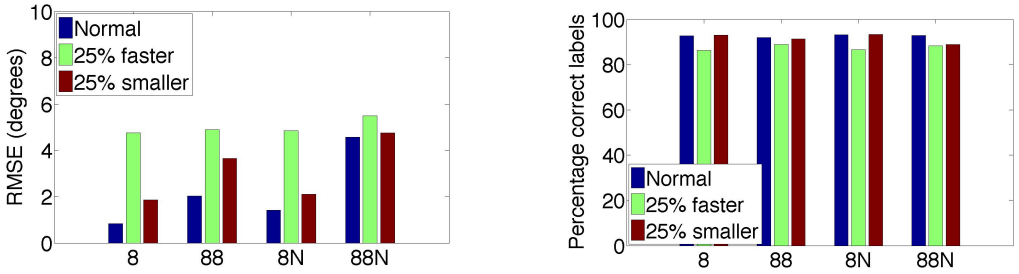


Fig. 6.6: *The baseline benchmarking condition. 20 second window of predicted and actual joint trajectories. Learning of horizontal figure-8 only, with no noise added. An exact trajectory given during prediction, with no hypothesis feedback. Only arm joint encoder input used to generate activation. Time in seconds on the x axes.*



(a) *Average RMSE.* ***(b)*** *Percentage correct labels.*

Fig. 6.7: *The benchmarking results for all trials, with hypothesis feedback. Average Root Mean Square Error (RMSE) over arm joint angles and average percentage correct labels. x axis notation: 8 - learning of horizontal figure 8 only, 88 - horizontal and vertical. N indicates noise added during learning. Color shading indicates prediction conditions.*

The hypothesis feedback had no discernible effect on the prediction of labels for the current experiment conditions.

6.3.3 Learning and Predicting Object Interaction

6.3.3.1 Introduction

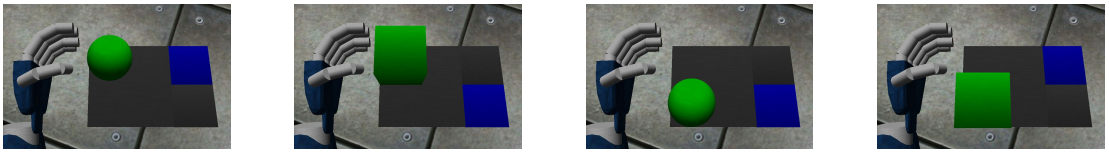
The goal of this experiment was to apply the approach to a set of robot tasks involving physical interaction with the world, and assess the ability to learn online from human demonstrations.

6.3.3.2 Method

Robot Embodiment The same robot embodiment as in the benchmarking trials was used, except that all 7 DOF of the left arm of the iCub robot were actuated. The hand of the robot was here teleoperated by an experienced human operator, the author. A 6 DOF rate-based SpaceNavigator input device was used, controlling the hand velocities in the hand frame. The neck was also actuated to make the robot fixate on the hand at all times, by implementing

the iCub gaze controller (iKinGazeCtrl). The neck pitch and yaw joint encoders were used as additional inputs to the NN.

Tasks Used The task setup consisted of a small planar table in front of the robot, and two different objects, a sphere (radius of 30 mm) or a cube (60x60x60 mm). See Fig. 6.3b. The hand of the robot started in the same initial position for each trial, while the object was initialized at rest in one of two positions, on the upper or lower part of the near end of the table. The goal was to push the given object to one of the two possible target locations on the far side of the table. See Fig. 6.8. A different hypothesis was used to represent each of the four tasks.



(a) *Task 1.* ***(b)*** *Task 2.* ***(c)*** *Task 3.* ***(d)*** *Task 4.*

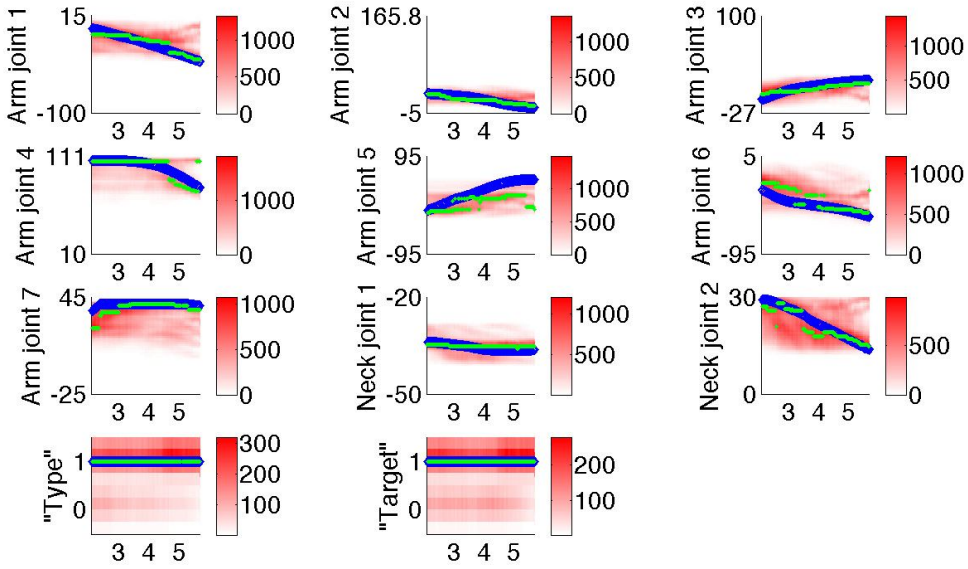
Fig. 6.8: *The 4 tasks used for the object interaction trials. Initial positions of hand and objects shown. Target location in blue.*

Neural Network Setup The NN was setup with input at two different delays (0.5 and 1 second). A total of 708 neurons for each delay was used, with 70 and 50 neurons per arm and neck joint, respectively. The full joint ranges for the arm joints were used, while the neck joint ranges were limited. The NN updated at approx. 7 Hz during learning and 40 Hz during prediction, on an 8-core Dell i7-2600 @ 3.4 GHz. The remaining settings were exactly as for the benchmarking trials.

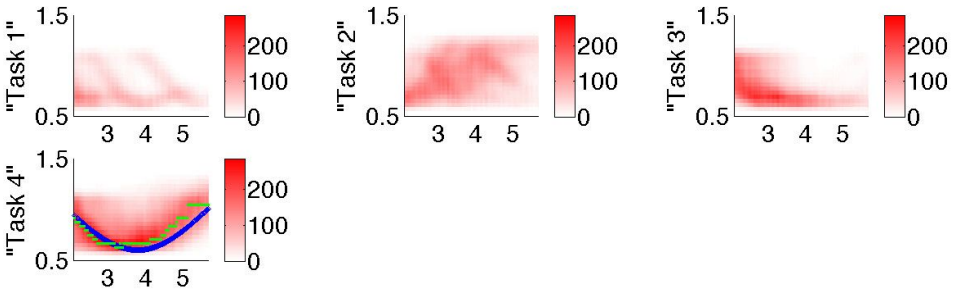
Experiment Procedure When the object position was within 15 mm of the center of the target position, the trial was deemed successful. A trial was deemed unsuccessful if the object fell off the table, or if more than 20 seconds passed. Unsuccessful trials were rerun, with the learning performed during this trial discarded, by reloading the set of NN weight matrices obtained before the trial started. 10 repetitions of each task was performed, in a mixed order, for a total of 40 trials. Both for learning and for prediction. Each trial took approximately 5-10 seconds to complete. Only arm and neck joint encoder input was used to generate activation during prediction.

6.3.3.3 Results

Fig. 6.9 shows an example comparison of the predicted and actual joint angles during observation of task 4. The prediction was performed at 0.5 seconds in the past. The activation across the neurons representing the values of the relevant sensor/hypothesis inputs is also shown. As can be seen multiple strong activations often exist. This information could potentially be used to give some estimate of the strength of the prediction, for example along the lines of the hypothesis feedback strength in section 6.3.1.5. A reasonable prediction of joint angles can be seen, with errors where there are multiple strong activations. The labels and the "Task" hypotheses are quite well predicted. As a comparison Fig. 6.10 shows an example of the prediction with activation only in the correct "Task" hypothesis. The prediction is very close to the actual followed during the trial, indicating that the hypothesis has been quite strongly grounded in the sensorimotor interaction during learning.



(a) *Arm/neck joint angles (in degrees) and labels.*



(b) *"Task" hypotheses.*

Fig. 6.9: *One attempt at task 4, moving a cube to the upper target. Neural activation (red intensity), predicted trajectory based on maximum activation (small green markers) and actual trajectory (large blue markers) shown. Only arm and neck joint encoder input used to generate activation. Time in seconds on the x-axes.*

Finally, Fig. 6.11 shows the complete results for predicting the labels and the 4 different "Task" hypotheses. The joint angles were not considered as there is not enough data for ground-truth trajectories on the 4 tasks used. The correct hypothesis was consistently predicted for task 4 (over 95% correct), while task 2 was the most inconsistent at under 60% correct. The prediction of the labels had less variation, with tasks 1, 2 and 4 at over 80% correct. More complete experiments are required to assess the true potential of the approach developed. The results obtained do indicate that learning and grounding can be performed online and provide reasonable prediction for non-trivial robot tasks however. A potential critique is that the labels and hypotheses used are representative of a more "strongly" designed approach, when compared to the self-organized parametric biases used by (Tani 2003). The latter method can for example produce novel behavior patterns by modulating the parametric biases arbitrarily. It might also be interesting to introduce novel tasks to the approach presented here, for example having the robot observe a sphere being pushed to a diagonal target position.

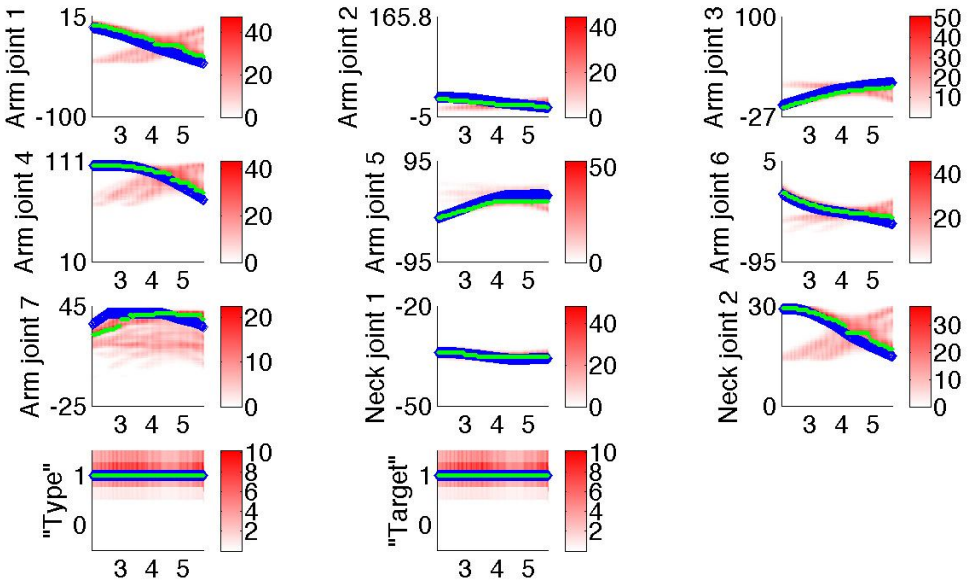


Fig. 6.10: *One attempt at task 4, moving a cube to the upper target. Only input from the correct "Task" hypothesis used to generate activation. Same coloration and notation as in Fig. 6.9.*

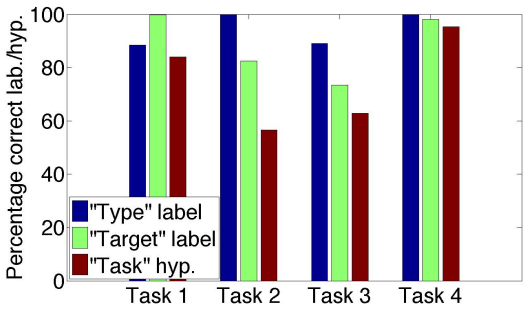


Fig. 6.11: *The object interaction results. Percentage correct labels (each with 2 levels), and percentage correct prediction of hypothesis (4 levels). Average over 10 executions. For the prediction of labels only data after which hypothesis feedback was available is used (after 4 seconds).*

6.3.4 Discussion

This section has described an approach for performing online learning of high-dimensional sensorimotor interaction, and the grounding of high-level labels and hypotheses in this interaction. A local Hebb-like learning rule was applied online across a set of time-delayed input layers. There are several shortcomings of the architecture. One is that there is no real mechanism for limiting the growth of the synapse weights. This is likely needed in any real-world application of the NN, where the learning rule parameters and interaction time and rate cannot be "tweaked" to each new situation. This is addressed in Section 6.4. Another potential shortcoming is the lack of a "hidden layer" or an equivalent functionality, where higher-level features of the sensor input can be established.

Benchmarking was performed on movements in the shape of a number 8, with different levels of noise and ambiguity. The results showed that the approach can learn and predict the future

of multiple movements without an explicit model of each one. Furthermore that task description labels and hypotheses can be grounded in the low-level sensor input during learning and later be activated by the same input. However, this task is quite far from any real-world task, and the results should be interpreted with this in mind.

A second imitation learning scenario with 4 goal-oriented pushing tasks was also explored. Here the approach showed a somewhat reduced ability to predict exactly, but also that the time-varying hypotheses had a strong grounding in the sensorimotor interaction. This enables prediction into the future by simply activating the relevant hypothesis. The approach can be of interest for predicting user intent, for example in assistive robotics. It may also be useful as the basis for more complex cognitive processes, such as mental simulation. However, the results are limited to prediction, and do not by themselves guarantee the suitability of the approach also for driving actuation. This will be explored in the next section.

6.4 Predictive Learning over Afferent and Efferent Signals

This section extends the previous by taking inspiration from the way children are able to learn while interacting with a teacher, in particular the use of prediction of the teacher actions to improve own learning. The architecture is based on two NNs that operate online, and in parallel, one for learning and one for prediction. A Hebbian learning rule is used to associate the high-dimensional afferent sensor input at different time-delays with the current efferent motor commands corresponding to the teacher demonstration. The predictions of future motor commands are used to limit the growth of the NN weights, and to enable the robot to smoothly continue movements the teacher has begun.

6.4.1 Neural Network Structure and Embedding

6.4.1.1 Overview

The two NNs and their interrelationships can be seen in Fig. 6.12. Each NN has multiple layers receiving the same afferent input, and each layer has a specific time-delay. The layers therefore represent the afferent input at different moments, from the present time t to T time steps into the past (denoted $t - T$). The afferent input includes sensor data from S sensors. Each sensor is represented by a set of neurons, here written as $\vec{s}$. The jth neuron for the afferent sensor input is denoted as a_j. High-level hypotheses about the task can also be provided, with the set of neurons representing one hypothesis written as $\vec{h}$. See Section 6.4.1.6 for more details. The jth neuron for the afferent hypothesis input is denoted as a_j^{hyp}. Each neuron in each set is assigned a specific sensor/hypothesis value, and the sets of neurons are therefore discretizations of the sensor/hypothesis inputs. Both NNs have one layer with neurons representing the efferent commands. For the work described here the arm joint velocities were assumed as commands, with the set of neurons representing each joint velocity being denoted as $\dot{\vec{\theta}}$. The ith neuron for the efferent commands is denoted as e_i.

6.4.1.2 Operating Modes

The NNs have two operating modes, learning and actuation. During learning, the teacher moves the robot through the task, while the learning NN learns the associations between the time-delayed afferent input and the efferent input representing the motor commands. The prediction NN simultaneously generates motor predictions. The activations generated by the prediction NN is used in the learning, and the prediction NN receives updates on the NN weights periodically.

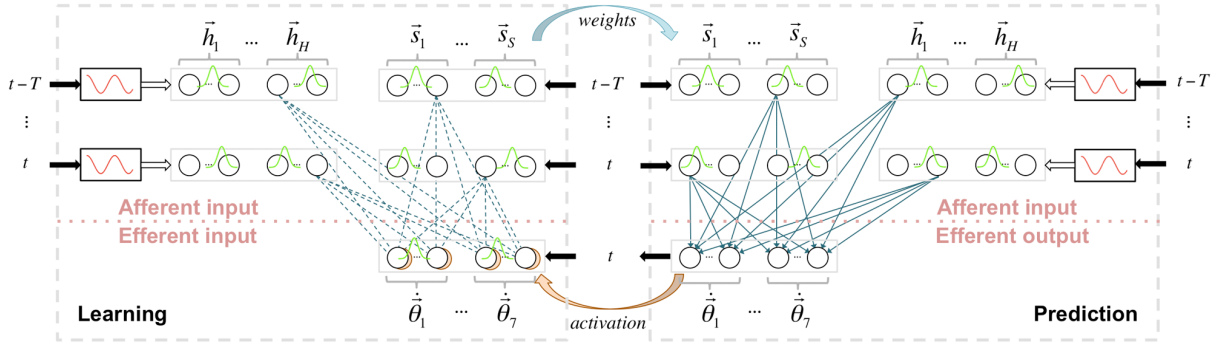


Fig. 6.12: *The two Neural Networks (NN). For clarity of presentation only the synapses for one neuron in each past layer are shown. Dashed synapses indicate learning (without propagation), solid arrow synapses indicate propagation of activity. Grey solid boxes represent layers, while the grey dashed boxes indicate the border of each NN. Thick solid arrows represent input from the outside. Example Gaussian activation curves shown in green.*

During actuation, only the prediction NN is active. It generates the motor commands to the robot based on the time-delayed afferent input received.

6.4.1.3 Afferent Input

The set of neurons representing the discretization of a sensor/hypothesis is activated according to a normalised Gaussian curve, with a mean at the actual sensor/hypothesis value and a set standard deviation. See Eq. (6.3), where χ is the value represented by the neuron, μ is the input value, and σ the standard deviation of the Gaussian curve used. Here σ was set equal to the separation in represented value of any two neurons for a given input. See also the Gaussian activation curves represented in Fig. 6.12.

$$v = f(\chi, \mu, \sigma) = exp\left(-\frac{(\chi - \mu)^2}{2\sigma^2}\right), \quad where\ v\ is\ a_j,\ a_j^{hyp}\ or\ e_i. \tag{6.3}$$

6.4.1.4 Efferent Input/Output

In the NN used for learning, the motor commands performed by the teacher is received as efferent input. The same normalised Gaussian activation curve as for the afferent input is used, see Eq. (6.3). In the NN used for prediction the neural activation in the afferent input layers is propagated to the efferent output, where the robot's own motor prediction and output is generated. Here the ith neuron is denoted as e_i^{pred}. As can be seen in Eq. (6.4), the activation of each of the n neurons in the efferent output layer (e_i^{pred}), is simply the sum of the m products of the activation of a given afferent input neuron (a_j or a_j^{hyp}) with its respective synapse weight ($w_{i,j}$). The value represented by the neuron with the maximum activation is taken as the respective joint velocity command.

$$e_i^{pred} = \sum_{j=1}^{m} w_{i,j} u, \quad where\ u\ is\ a_j\ or\ a_j^{hyp}. \tag{6.4}$$

6.4.1.5 Predictive Hebbian Association

In the NN for learning no activation is propagated and a prediction-based learning rule is applied across all synapses. See Eq. (6.5). The change of a given synapse weight each iteration ($\Delta w_{i,j}$) is proportional to the activation of the respective afferent input neuron (a_j or a_j^{hyp}) and the difference in actual and predicted activation of the efferent command neuron. The rule is a derivative of the Hebbian (Hebb 2005) learning of the previous section, but uses the predictions made in parallel to attempt to limit the synapse weights. The rule superficially resembles that of Restricted Boltzmann Machines (RBM) (Smolensky 1986), but there are anyhow differences, for example no hidden layer is here used. Rather the learning occurs with the actuation-driven neural activation representing the afferent input at different points in the past.

$$\Delta w_{i,j} = \eta y \left(e_i - e_i^{pred} \right), \quad where\ y\ is\ a_j\ or\ a_j^{hyp}. \tag{6.5}$$

6.4.1.6 High-Level Hypothesis Labels

High-level input can be provided to the NNs through the hypothesis labels. An example could be a specific text label describing the task, like "pick up cup", that the teacher gives to the robot during learning. Each hypothesis input is a binary value (e.g. the existence, or not, of a given label), but the representation is spread over a set of neurons. A pattern generator is used to give a time-variable pattern to the neural activation. A sinusoidal pattern was selected for the implementation presented here, as seen in Eq. (6.6). Here τ is the desired period of the pattern (here 60 seconds). In the learning NN the activation of a hypothesis will thus continuously activate different neurons, the activation of which are associated with the current motor commands. The activation of a hypothesis during robot actuation can then be used to impose a given trajectory of motor commands.

$$\mu^{hyp} = \lambda cos(2\pi t/\tau),\ where: \quad \lambda = \begin{cases} 1 & with\ hypothesis, \\ 0 & otherwise. \end{cases} \tag{6.6}$$

6.4.1.7 Embedding in a Developmental Robot Platform

The overall architecture for embedding the NN in the real-time control loop of a robot an be seen in Fig. 6.13. The YARP (Metta et al. 2006) communication protocol was again used. An actuation manager directly controls the joint velocities of the left arm of the iCub humanoid robot ($\dot{\vec{q}}_{command}$), based on either teacher input ($\dot{\vec{q}}_{teacher}$) or the predictions of the NN ($\dot{\vec{q}}_{NN}$). A separate NN performs the learning simultaneously, and updates the weights used for prediction periodically. Both NNs receive input from the same set of sensors, from the present time and from given times in the past. This is achieved through a delay queue. The gaze of the iCub is made to follow the left hand independently. Examples of learned NN weights can be seen in Fig. 6.22 for joint encoders, and Fig. 6.26 for visual sensors.

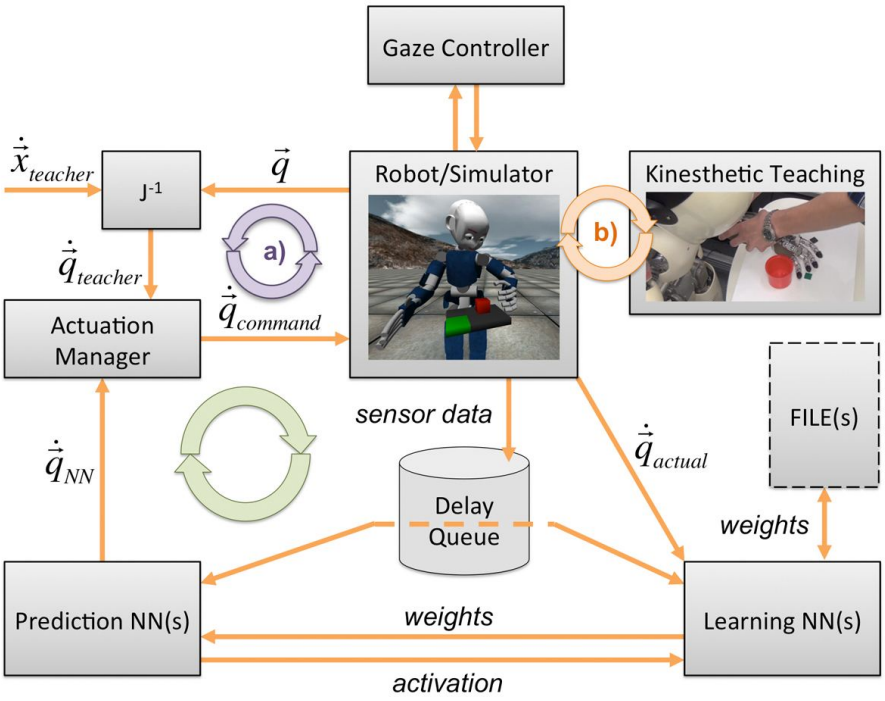


Fig. 6.13: *Overall architecture and embedding in the sensorimotor coordination loop. Large green circular arrow indicates the robot actuation loop, small circular arrows indicate the two possible teacher actuation loops: a) teaching a physical or simulated robot through a user interface generating Cartesian velocities (e.g. a joystick), and b) performing kinesthetic teaching on a physical robot (physically moving the robot). J indicates the Jacobian.*

6.4.2 Simulated Robot Experiment

6.4.2.1 Introduction

The experiment was performed in the iCub simulator (Tikhanoff et al. 2008). The task setup consisted of a small planar table in front of the robot, and a small red cube (40x40x40 mm). See Fig. 6.14 for the table used, Fig. 6.15 for the general setup. The goal of each trial was to push the red cube to one of the two possible target locations on the far side of the table, shown in green. The robot was to learn to take over control from the teacher after a preset time into a demonstration (2 or 4 seconds) and complete the task. The main performance measure was the percentage of successfully completed attempts. The hand of the robot started in the same initial position for each trial, while the red cube was initialized according to a random set of positions within a predefined area. Two different square areas were used for the random distribution, with the size (denoted as d) being 40 or 20 mm. In Fig. 6.14 dashed black lines indicate the borders for the center of the cube, and the grey areas the outer edges.

6.4.2.2 Method

6.4.2.3 Teacher Interaction

The teacher used a 6 DOF 3DConnexion SpaceNavigator input device to control the Cartesian velocities of the left hand ($\dot{\vec{x}}_{teacher}$ in Fig. 6.13). The iKinGazeCtrl gaze controller (Pattacini 2010) was used to make the robot track the position of the left hand of the robot with the eyes and head. The teacher was provided with a graphical representation of the average RMSE over

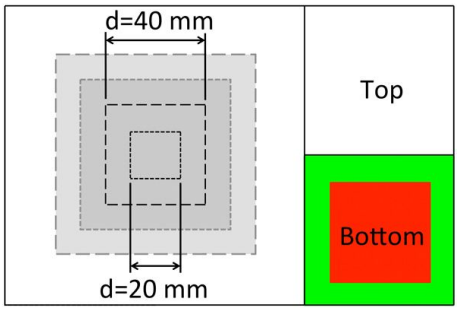


Fig. 6.14: *The "object-pushing" task.*

the 6 predicted joint velocities, see Fig. 6.15. This provided a tool for the teacher to assess how well the robot was able to predict the current movement performed.

Fig. 6.15: *View given to teacher during testing. Timer in top-right corner, visualisation of average Root Mean Square Error (RMSE) over joint velocities in bottom-right corner.*

6.4.2.4 Sensor Input

Sensor data delayed by 0, 1.5 and 3 seconds was fed to the NNs. The sensors used were: a) The 6 joint encoders of the left arm, b) the 2 joint encoders of the neck, and c) 54 vision sensors (18 red, 18 green, and 18 blue pixel counters, see Fig. 6.27). See Table 6.1 for the assumed resolution (the number of neurons used), min and max values. Each hypothesis was represented by 35 neurons.

6.4.2.5 Robot Actuators

The actuators used were 6 of the left arm motors (joint velocities commanded), see Table 6.2. Shoulder yaw was kept at 7 degrees. The robot actuation was stopped when the average maximum neural activation across the 6 joint velocities fell below 0.1. The system was tested on an 8-core Dell i7-2600 @ 3.4 GHz. Both the learning and prediction were split over separate NN modules, by running two instances with half the joint velocities for each. The learning achieved 10-15 Hz and the prediction 30-35 Hz.

Tab. 6.1: *Specifications of Sensors Used in Experiment.*

		Resol.	Min.	Max.	Units
Arm	Shoulder pitch	35	-100	15	*degrees*
	Shoulder roll	35	-5	166	
	Elbow	35	10	111	
	Wrist pronosupination	35	-95	95	
	Wrist pitch	35	-95	5	
	Wrist yaw	35	-25	45	
Neck	Pitch	35	-65	-20	*degrees*
	Yaw	35	-5	40	
Vision	24 small pixel counters	20	120	1600	*pixels*
	30 large pixel counters	20	400	3200	

Tab. 6.2: *Specifications of Actuators Used in Experiment (in deg/s).*

		Resolution	Min.	Max.
Arm velocity	Shoulder pitch	45	-15	15
	Shoulder roll	45	-15	15
	Elbow	45	-15	15
	Wrist pronosupination	33	-15	15
	Wrist pitch	33	-15	15
	Wrist yaw	33	-20	20

6.4.2.6 Experiment Procedure

The main experiment conditions can be seen in Table 6.3. A total of about 10 hours of testing was performed, with the author as the teacher. For training sessions, the teacher could switch between own and robot actuation with a button. 40 training attempts were performed. The robot only learned during periods of teacher actuation, and only from successful attempts. For testing sessions the teacher started actuating, and the robot automatically took over after a preset time had passed (2 or 4 seconds). 40 attempts were used for measuring performance for each experiment condition. A separate hypothesis for each target location was given on conditions 3, 4, 7 and 8.

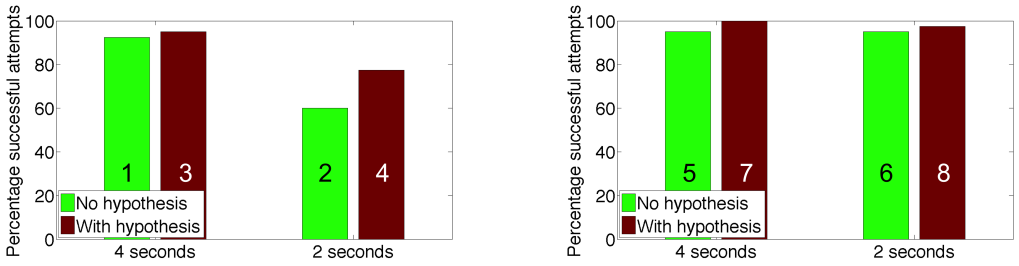
Tab. 6.3: *Main Conditions for Experiment.*

		d=40 mm	d=20 mm
No hypothesis	4 sec.	Condition 1	Condition 5
	2 sec.	Condition 2	Condition 6
With hypothesis	4 sec.	Condition 3	Condition 7
	2 sec.	Condition 4	Condition 8

6.4.3 Main Results

6.4.3.1 Percentage of Successfully Completed Attempts

The percentage of successfully completed attempts for the main experiment conditions can be seen in Fig. 6.16. There was a high success rate for most conditions, but the performance dropped sharply when both a high dispersion of the initial locations of the cube (d=40 mm) and a short amount of time before beginning actuation (2 seconds) were used. Providing hypotheses on the given target locations improved performance by 29.2% for this situation, and had a positive effect in general.



(a) *d=40 mm.* ***(b)*** *d=20 mm.*

Fig. 6.16: *Percentage of successfully completed attempts for the main experiment conditions. With or without hypothesis about task to perform, and beginning actuation after 4 or 2 seconds. All results with 40 attempts for training. Condition number indicated on each column.*

6.4.3.2 Trajectories Followed

The differences in the joint space trajectories followed for the two tasks are shown in Fig. 6.17. It can be seen that the trajectories for the two tasks are quite similar in shoulder roll,

while for other joints they differ partially (e.g. shoulder pitch and elbow) or completely (wrist pitch). The different start locations of the red cube also required quite different trajectories in some joints (e.g. wrist yaw for the top target). In Fig. 6.18 the corresponding neck angles are shown. The trajectories can mainly be distinguished in the neck pitch joint.

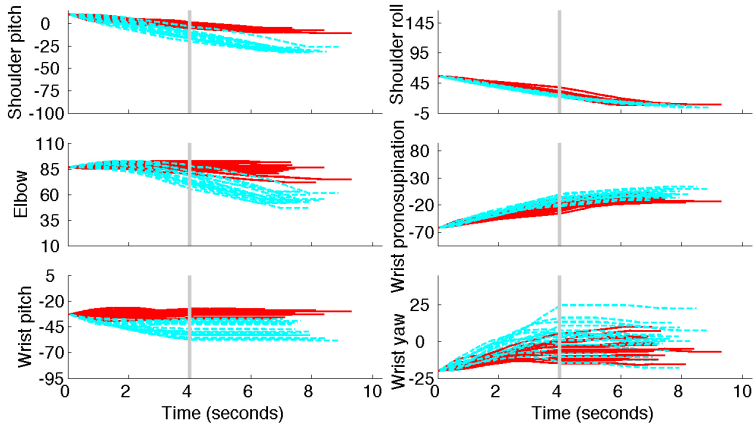


Fig. 6.17: *Arm joint angle trajectories for actuation from 4 seconds (no hypothesis given and d=40 mm; condition 1), for all successful attempts. Both bottom target (solid red lines) and top target (dashed cyan lines). Grey vertical line indicates start of robot actuation.*

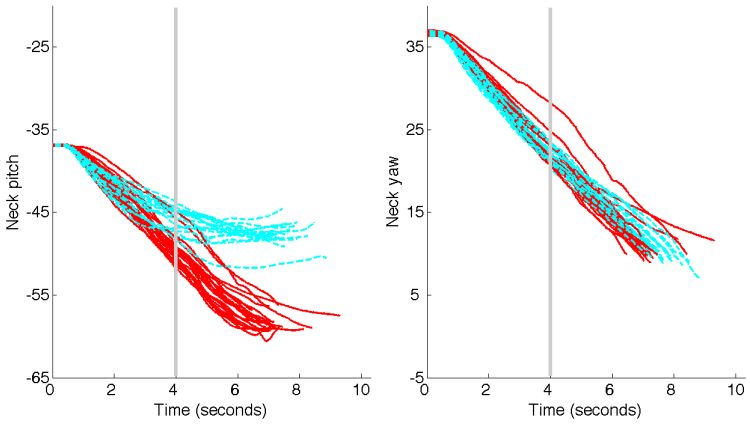
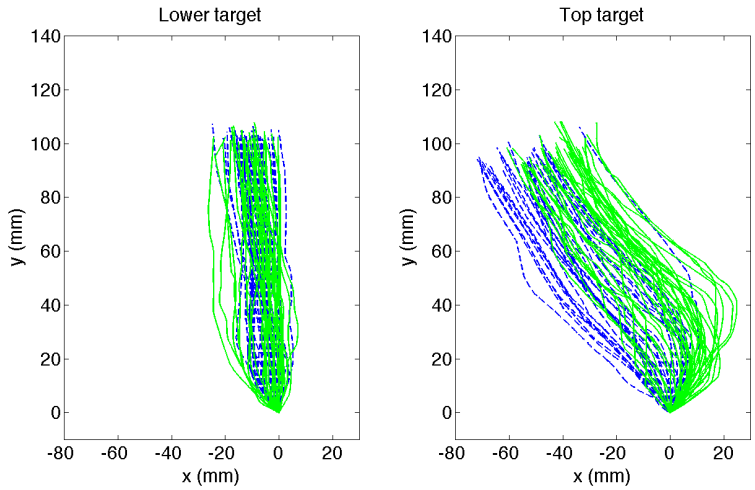


Fig. 6.18: *Neck pitch and yaw angle trajectories for condition 1 (see Fig. 6.17).*

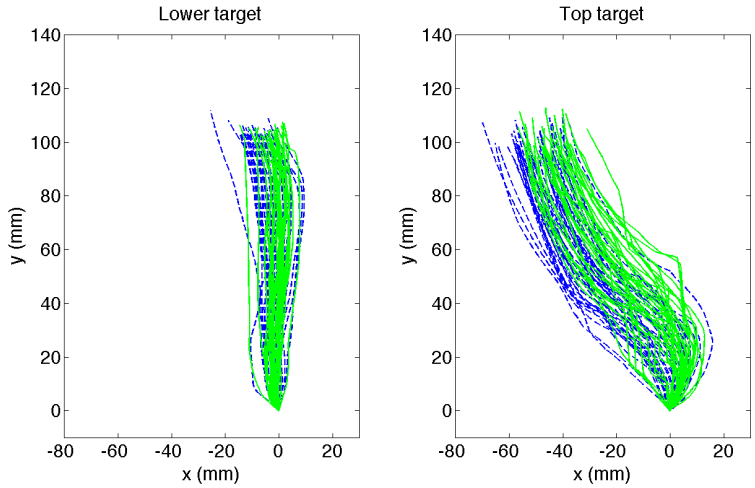
Fig. 6.19 shows the Cartesian trajectories of the hand in the horizontal plane. It can be seen that many of the trajectories for the two targets cross, making the task of distinguishing one from the other more difficult. The large dispersion of final positions follows from one of the characteristics of the tasks performed, the ability to push the cube with any part of the hand/wrist/fingers.

6.4.3.3 Development of Neural Network Weights

Limiting the synapse weights in Hebbian learning is often addressed through Oja's rule (Oja 1982) or similar approaches for "forgetting". For a NN that is directly embedded in the real-time loop of a robot agent, the weights will already be somewhat limited. That is, the robot will only be interacting with the world for a given period of time and at a given rate (10-20 Hz here).



(a) *d=40 mm.*

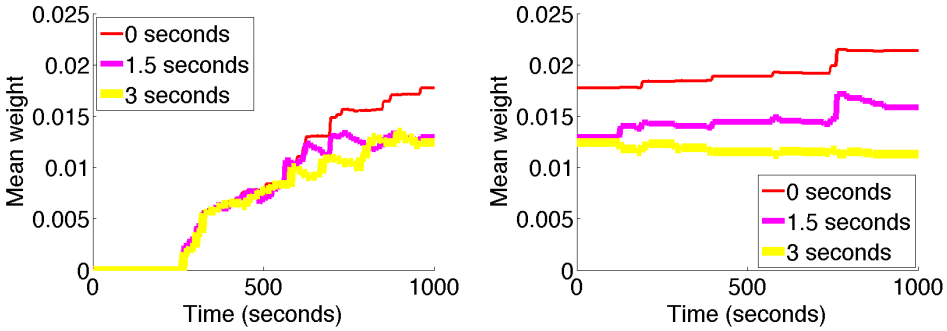


(b) *d=20 mm.*

Fig. 6.19: *Cartesian trajectories of the hand in the $x-y$ plane (top view) for learning (solid green lines) and actuation (dashed blue lines) trials.*

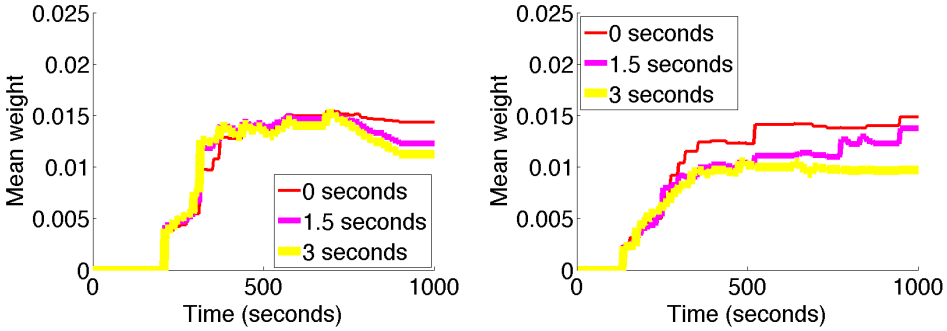
However, some neural mechanism for normalizing the weights is still desirable. The error in the prediction is here used, which means the NN mainly "forgets" (and learns) when the efferent prediction is far from the efferent command. That is, both learning and "forgetting" depend on the performance in the current context.

As can be seen in Fig. 6.20a the NN weights increased sharply at first, but had a gradual reduction in slope towards the end of the 40 training attempts. There was little overall increase in the mean weights when running an additional 40 training attempts after the experiment, as can be seen in Fig. 6.20b. However, there was a sudden increase after about 750 seconds that likely stems from variability in the teacher demonstrations. Similar effects were also seen in the other conditions. Overall it seems the training used was sufficient to stabilise the weights for the tasks performed. Fig. 6.21a shows that a less variable initial distribution of positions (d=20 mm) made the tasks easier to learn, with the mean weights quickly reaching a high level. Fig. 6.21b shows the development with d=40 mm and hypothesis given.



(a) Original 40 training attempts. *(b) Additional 40 training attempts.*

Fig. 6.20: *Development of the mean of the Neural Network (NN) weights for no hypothesis given and d=40 mm (conditions 1 and 2). Over the original 40 attempts and after 40 additional attempts. Mean shown individually for 3 delays used.*



(a) With d=20 mm. *(b) With hypothesis given.*

Fig. 6.21: *Effect on the development of the mean of the Neural Network (NN) weights when: a) d=20 mm (conditions 5 and 6), and b) when hypothesis is given (conditions 3 and 4). Mean shown individually for 3 delays used.*

6.4.3.4 Final Neural Network Weights

Examples of the final NN weights learned can be seen in Fig. 6.22 and Fig. 6.23, for delays of 0 and 3 seconds, respectively. For most sensors there are multimodal distributions of weights to the possible motor actions, suggesting non-trivial relationships between sensing and actuation. It can be seen that the weights at the two delays have observable differences, especially in the neck joints, even though only 3 seconds separate the two sets of weights. The NN architecture presented takes advantage of such additional information by basing the actuation on the recent past, not only the current sensor input.

6.4.4 Sensitivity Analysis

The effect of three changes to the sensorimotor dynamics were explored. All three changes were applied to experiment condition 1 (actuation from 4 seconds, no hypothesis given and d=40mm).

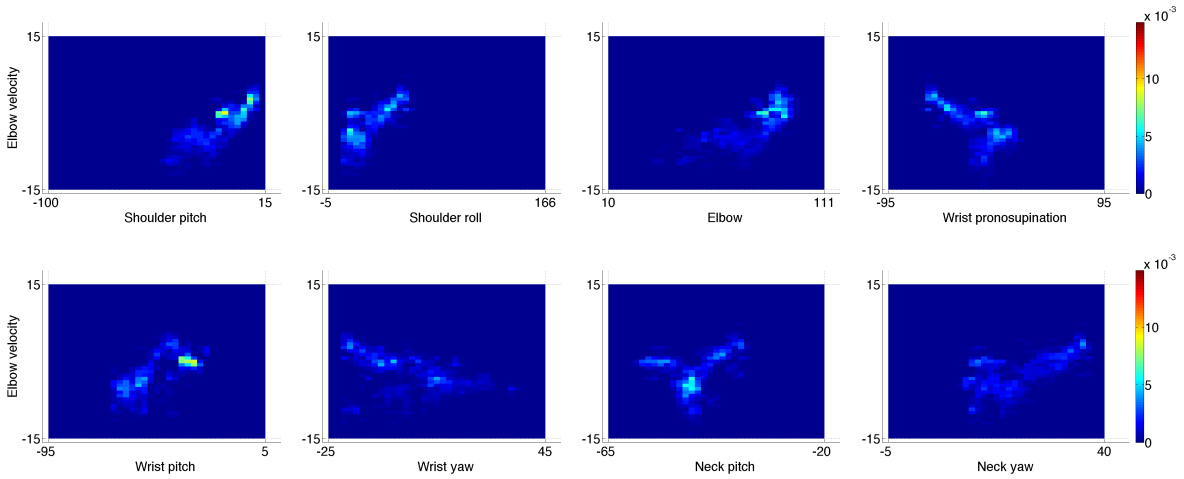


Fig. 6.22: *Example learned Neural Network (NN) weights from arm and neck encoders to elbow joint velocity, at delay of 0 seconds.*

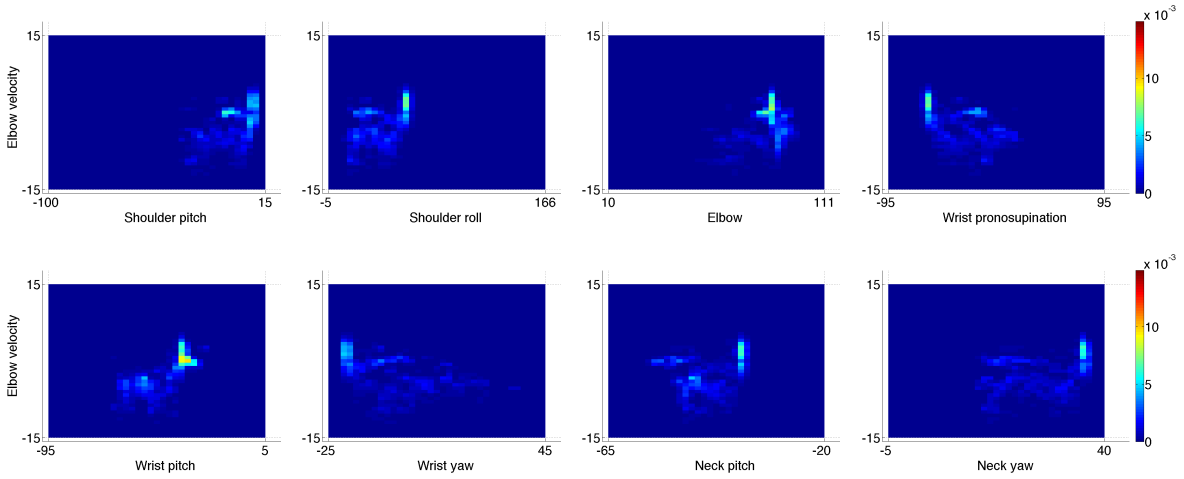


Fig. 6.23: *Example learned Neural Network (NN) weights from arm and neck encoders to elbow joint velocity, at delay of 3 seconds.*

6.4.4.1 Learning an Additional Task

The first effect explored was the addition of a different task, touching one of two static spheres, hereby denoted as the "two-button" task. See Fig. 6.24. One of the spheres were green, the other red, with the order assigned randomly. The goal of the task was to push with the palm against the green sphere. The NN weights obtained during training on condition 1 was first loaded. The NNs were then trained on 40 attempts on the "two-button" task, and was tested to confirm 100% successful execution after 4 and 2 seconds. The robot was then tested on the original task, and was able to complete 82.1% of the attempts. This compared to the 92.5% before learning the additional task, a reduction of 11.2%.

6.4.4.2 Random Visual Effect

Condition 1 was rerun with a random visual effect, a red sphere placed in a randomised location within a 400 mm square on the floor. See Fig. 6.25 for three examples. The visual

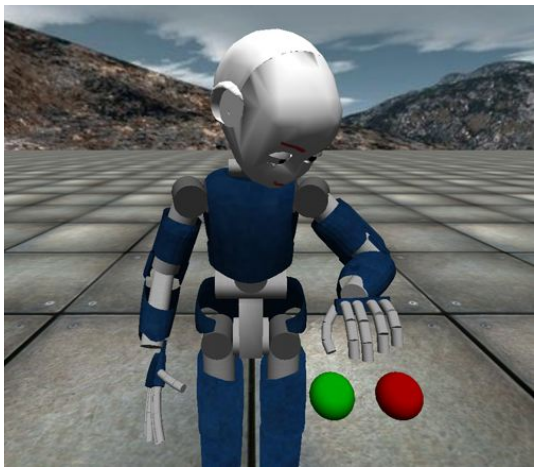
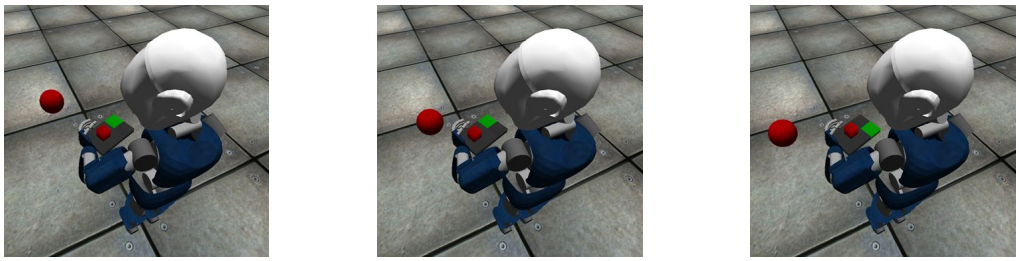


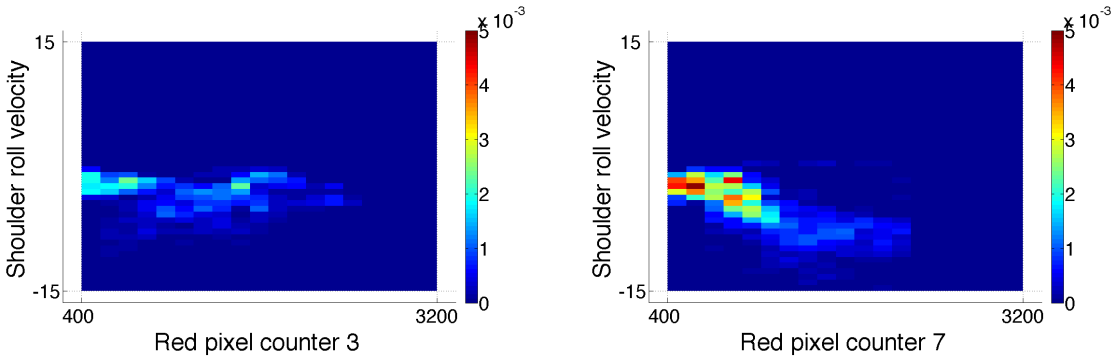
Fig. 6.24: *"Two-button" task.*

impact was approximately the same as the smaller red cube on the table, but occurred mainly in the top half of the robot's retina. The robot was trained with 40 attempts as before. During testing the same success rate as without the random effect was achieved, that is 92.5%. The corresponding NN weights for the areas of the retina affected by the random sphere were also lower and more even that those affected by the red cube, see the example in Fig. 6.26. The sensors with the highest respective weights are shown, sensors 3 and 7 (see Fig. 6.27a).



(a) ***(b)*** ***(c)***

Fig. 6.25: *Including a red sphere in a random location in front of the robot.*

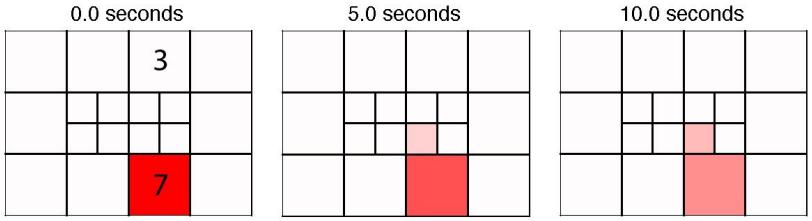


(a) *Sensor 3.* ***(b)*** *Sensor 7.*

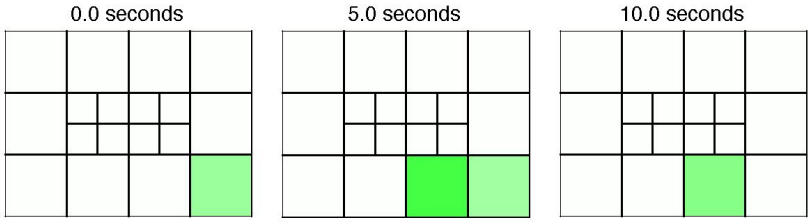
Fig. 6.26: *Example Neural Network (NN) weights from red pixel counters 3 and 7 to shoulder roll velocity. See Fig. 6.27a for a visualization of the sensors concerned.*

6.4.4.3 Using Vision Sensing Only

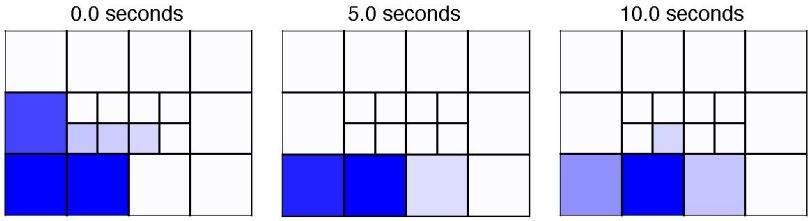
Condition 1 was rerun with only vision input, with 40 attempts for training as before. See Fig. 6.27 for example visualisations of the visual information received, and Fig. 6.28 for the corresponding situations. The robot was able to complete 90% of the tasks after 4 seconds without hypothesis given (condition 1). The reduction in sensors led to faster execution of the NNs, up to 13-16 Hz for the learning and 37-40 Hz for the prediction.



(a) Red pixel counts.



(b) Green pixel counts.



(c) Blue pixel counts.

Fig. 6.27: *Visualizations of visual input at three times for one attempt.*

(a) 0.0 seconds. *(b) 5.0 seconds.* *(c) 10.1 seconds.*

Fig. 6.28: *Actual situations corresponding with Fig. 6.27.*

6.4.5 Discussion

The tasks used here are quite simple and do not require great precision. However, they have a high degree of overlap and are performed in a small part of the workspace. Tasks involving a

greater sensor data diversity may be easier to learn, at least when the additional data is correlated to the task. Humans are also good at taking advantage of the richness of the context to drive action. Remembering the pin code for a bank card is often much easier in the right context, i.e. with your fingers on the cash machine keypad. In fact, both motor memory formation and decay has been found to be strongly context dependent (Ingram et al. 2013).

The learning could also have been "hard-coded" to improve performance. First, a separate set of NN weights could have been used for each target, or even for each approximate start location. That is, each movement "class" could have been modelled and then represented in code by a symbol, and only activated when certain predefined conditions were satisfied. Second, estimating the 3D pose of the specific cube of interest with more elaborate vision algorithms is certainly feasible, but would require a second mechanism for estimating from the context when the 3D pose of the red cube is of interest, and when not. Reducing the need for explicitly including such gate-selection mechanisms (Yamashita & Tani 2008) may help circumvent deeper issues like the symbol grounding problem (see (Harnad 1990)).

Finally, the robot here learned a direct association between the sensor input and the joint velocities. Thus no implicit knowledge of the kinematics of the robot was assumed, although this could likely help speed up the learning.

6.5 First Implementation on the Physical iCub Robot

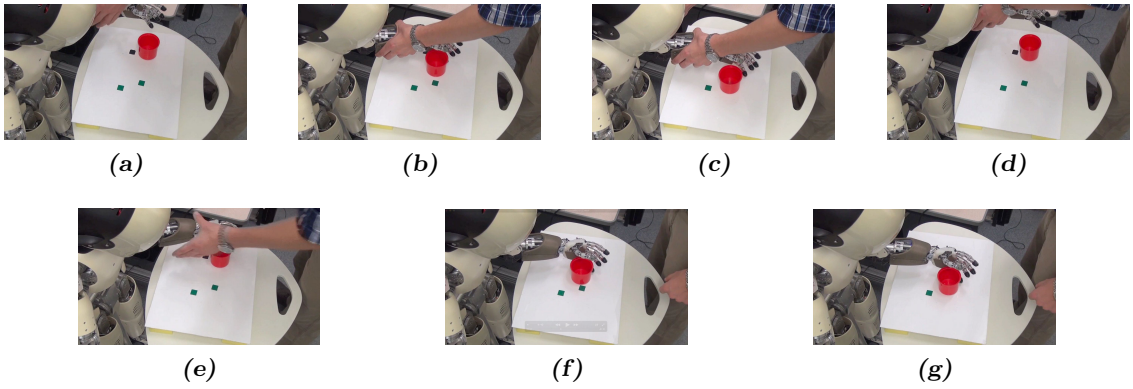


Fig. 6.29: *First testing on the real iCub platform. a) to c): complete user demonstration; d) and e): user starting task; f) and g): robot completing task.*

Two simple tests of the ability to learn and actuate on the physical iCub robot were performed. Kinesthetic teaching was used, where the teacher directly moved the joints of the robot. See teacher actuation loop b) in Fig. 6.13. For the first test, the task was to move a red cup from one of two points on a table to one of two targets to the right. See Fig. 6.29. The input to the NN was the 7 left arm encoders and the yaw neck encoder. The head tracked the red cup using a colour segmentation algorithm, see (Morse 2010). A different hypothesis was given to each of the four possible tasks and used during actuation. Only one kinesthetic demonstration was given for each task. For the second test, no hypothesis was given, and the robot was shown one task, moving the cup from one point to another. The teacher started the test, then grabbed the cup and slowly moved it to the final point. As can be seen in Fig. 6.30, the robot followed the cup with the hand, showing a simple (task-limited) sensorimotor coordination.

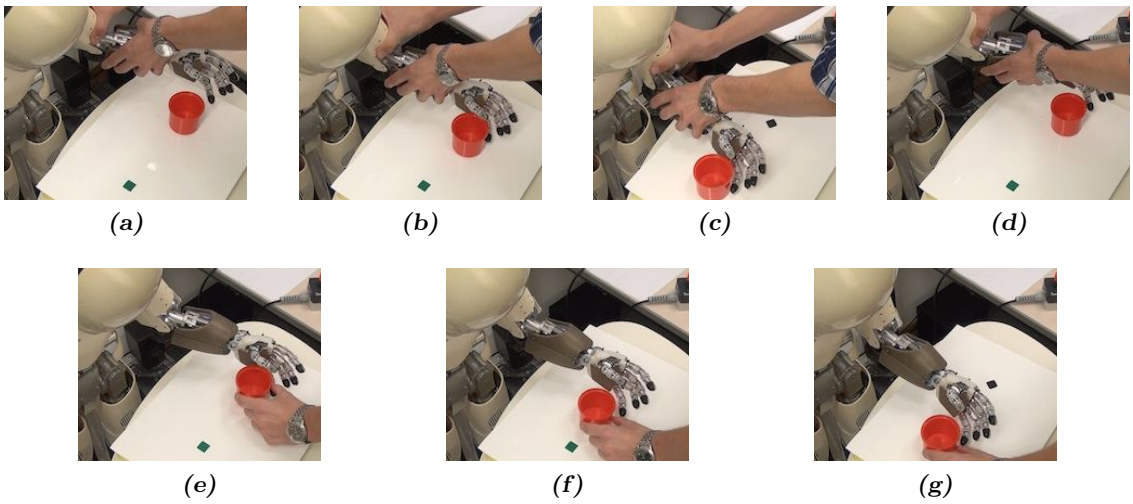


Fig. 6.30: *Demonstrating simple sensorimotor coordination. a) to c): complete user demonstration; d) user starting task; e) to g): robot following cup.*

6.6 Discussion and Conclusion

A robot that is to take inspiration from how children learn and interact with their environment should likely have, at least, the following capabilities:

1. Automatically extract task-relevant information from the robot's full high-dimensional sensory input.
2. Predict future teacher actions from the observation of the dynamics of the relevant sensory input.
3. Transition smoothly from teacher-guided action to own action based on such predictions.
4. Integrate information from high-level labels with the low-level prediction, when available.
5. Perform both learning and prediction online while the human teacher is providing demonstrations.

The work presented here is an attempt to move in the general direction of such capabilities, but so far has several important limitations. Especially when being applied to assistive manipulators, and being used in a shared control scenario. Here the user demonstrations may be inaccurate, and perhaps vary greatly from attempt to attempt. The direct mapping from sensory input to actuation may not be able to extract the complex underlying features of such demonstrations. The actuation of the robot may similarly get "lost" in a pose where the user has never taken the robot, even though it may seem obvious to the user how to complete the task. That is, the behaviour should perhaps be more "goal-directed"?

The next Chapter will outline some ideas in this direction, by allowing the robot to automatically take over on well-predicted tasks, and by letting the sensory input drive the activation of multiple competing sensorimotor primitives. Including "homing" or "goal-directed" behaviours.

7. HELPING THE USER COMPLETE TASKS

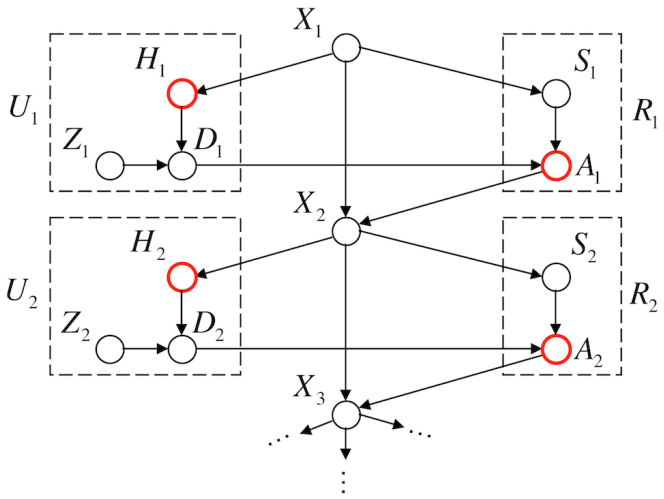


Fig. 7.1: *Directed acyclic graph representing the problem approached in this Chapter: Making the robot actuation A follow the intent of the human user H as much as possible, in effect helping the user complete tasks with the assistive manipulator. Subscripts indicate time-steps, see Section 2.2.3 for details.*

7.1 Introduction

The last three Chapters have outlined two different ideas for adapting the use of sensors in the robot to help the user. For the collision-limitation behaviour of the SDAC concept, improvements in performance were found on typical tasks in a household environment in virtual experiments. For the interactive learning of tasks in a humanoid robot, the ability to learn and predict simple object manipulation tasks was demonstrated. This chapter will attempt to bring the two approaches together, as a step towards achieving a system that can help the user complete real-world tasks on actual assistive manipulator platforms. The goal is to bring about a higher user satisfaction, a higher performance, and a lower user workload. The chapter concerns work in progress, and therefore includes few tangible results. Hopefully the discussion of the ideas will help indicate possible directions the work can take in the near future.

7.2 Automatic Switching from User Control to Robot Control?

7.2.1 Introduction

If an assistive manipulator is to help the user complete tasks, then it may be beneficial for the robot to automatically provide assistance when needed. This can help to: a) make the interaction more fluid and natural, b) reduce the need for additional input from the user on when to make

the switch, and c) give the user immediate feedback on when the robot's predictions are good, which can potentially help reduce the amount of demonstrations required to a minimum. This section will outline the ongoing work in this direction, including different criteria that could be used to drive the switching, a first implementation, and first results on the simulated iCub robot.

7.2.2 Potential Criteria For Switching

To provide assistance when needed, the robot needs a good estimate of the "need" of the user. At least a quantitative criteria for when it's own predictions are reliable, preferably one that leads to switching that is obvious and transparent to the user. Three criteria were considered in this preliminary work:

1. *Error in prediction.* The NNs continually produce predictions of motor commands and these can be compared with the actual commands produced by the teacher demonstrations. For example taking the RMSE of the Cartesian velocity components ($\dot{x}_1$, $\dot{x}_2$, ... , $\dot{x}_6$) at each point in time.

2. *Mean neural activation.* The activation in the neurons representing the motor commands can likely also provide useful information. A high mean activation could indicate that the recent input combination is similar to one previously learned.

3. *Entropy of neural distributions.* The type of distribution in the output neurons is another option. A uniform distribution (uniform activation in all neurons representing a given motor) would indicate that no particular action is predicted. The entropy of such a distribution would be high, as compared to one where one or more specific actions have a higher activation. This could be useful information when switching behaviours.

7.2.3 First Implementation

The architecture used for the first implementation can be seen in Fig. 7.2. The motor commands used are the Cartesian velocities of the iCub arm, which are resolved into joint velocities using the known kinematic structure of the iCub left arm. This should make the commands more relevant to physical manipulation tasks. The actuation manager controls the switching between teacher and robot control, here based on the error in prediction and the mean neural activation.

In teacher control mode, the actuation manager compares the predicted Cartesian velocities with those generated by the teacher at each point in time. The RMSE was used. When the RMSE falls below a given threshold, the robot is given control. During robot control the NN predictions are used to drive the actuation of the robot arm. If the mean activation falls below a given threshold, the arm is stopped. This provides some safety in the system, and gives the user time to take over control (by for example pushing a button). The NN architecture used can be seen in Fig. 7.3. As can be seen from the figure, no high-level labels or hypotheses were used for this initial implementation.

The sensory information received by the NN can be seen in Fig. 7.4. In all 69 distinct sensor readings were received in real-time. Both the learning and prediction NN ran at approximately 30 Hz on an 8-core Dell i7-2600 @ 3.4 GHz. One of the keys to the fast execution was an optimised C++ code. Among the improvements to the code was to selectively update only the NN weights from neurons with a non-zero (below threshold) activity.

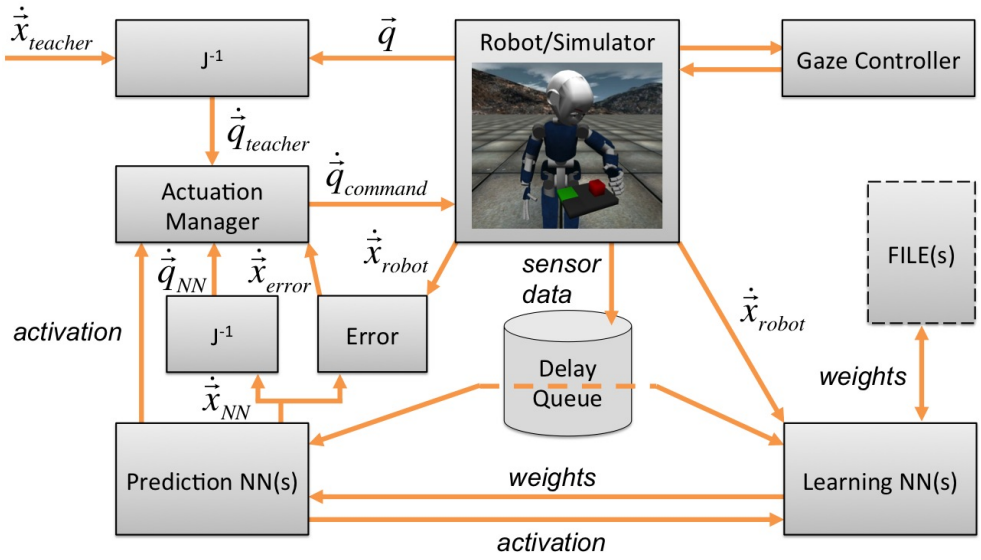


Fig. 7.2: *Architecture for first implementation of automatic switching between robot and user/teacher. J indicates the Jacobian.*

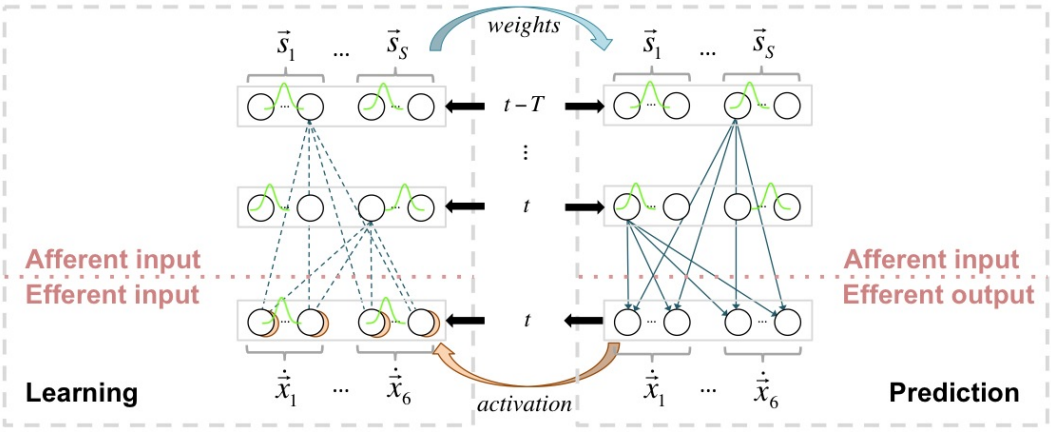


Fig. 7.3: *The learning and prediction Neural Networks (NN) used.*

7.2.4 First Experimental Setup

A first experimental setup for evaluating the automatic switching was created. As in the previous experiments, the tasks involved pushing objects on tables. Here two additional tasks were added, both based on a narrow path that the robot should learn to push the object along. See Fig. 7.5. The new tasks (tasks 3 and 4 in the figure) also had very different color coding. For example a green cube was to be pushed (previously the target color) along a red table (previously the color of the cube) to a grey target (previously the color of the target). This, and the narrower tolerance on the movements to avoid the cube falling off, added some variation to the task set.

However, before generating substantial experimental data with this implementation and experimental setup, there is still an open question as to how to compare performance. The main objective should be to have the robot take over as much of the actuation as possible. Thus the percentage of actuation performed by the robot during the trials could be a good metric for performance. However, the success of the attempts still has to be taken into account. For example, situations where the robot actuates most of the time but is not completing the tasks

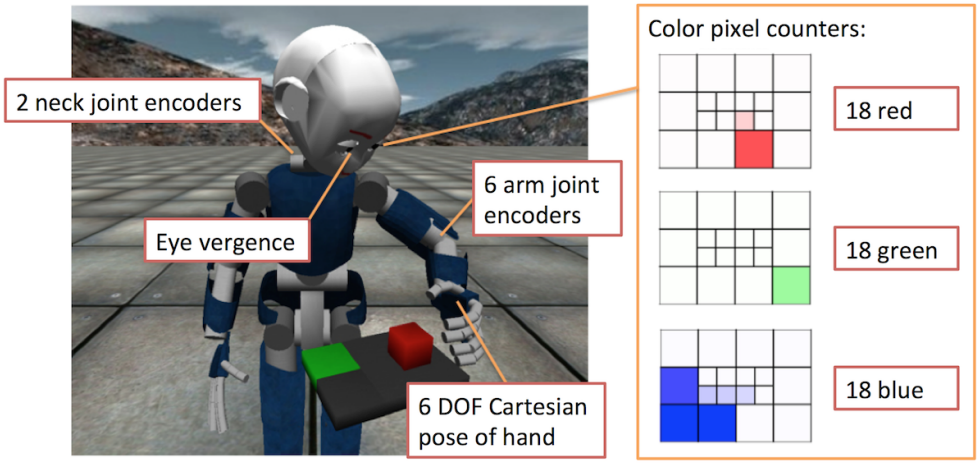
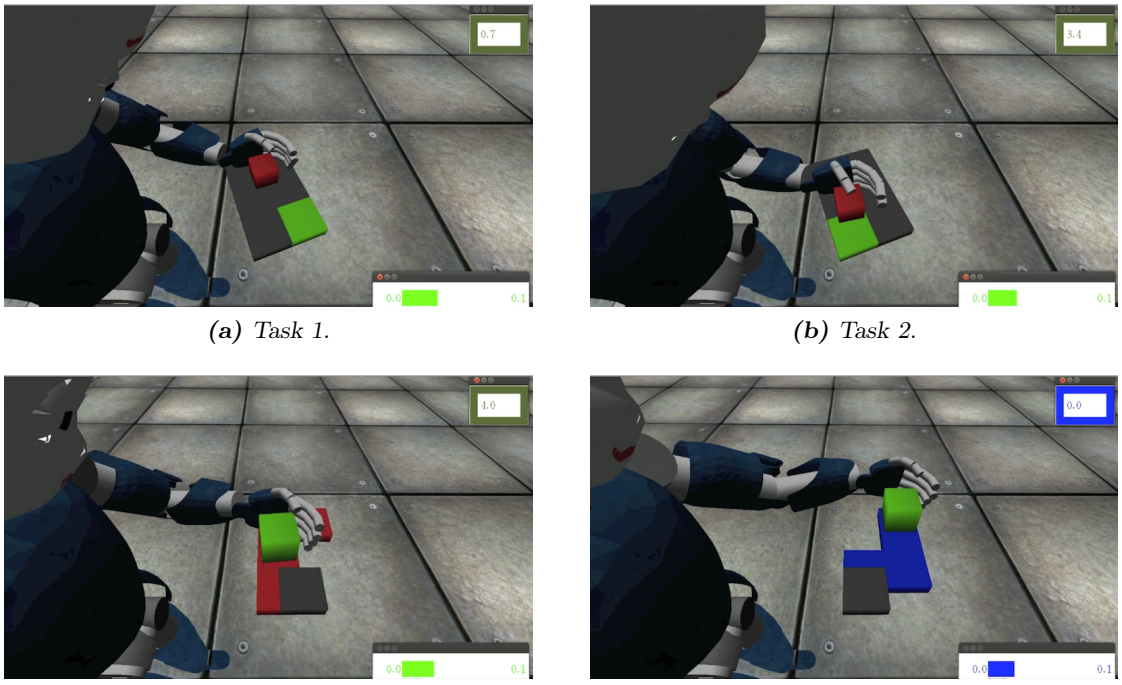


Fig. 7.4: *The sensory context used for the first implementation and testing.*



(a) *Task 1.* ***(b)*** *Task 2.*

(c) *Task 3.* ***(d)*** *Task 4.*

Fig. 7.5: *The four tasks attempted in the first testing of automatic switching.*

should be avoided. One way to approach this issue is to only count "successful" attempts, in this case attempts where the cube ended up in a resting pose on the target indicated. Or to require a given ratio of success, for example 90%, for including a session in comparisons between conditions. But the time T taken to get to a stable percentage robot actuation R is also of interest. See the case of two hypothetical conditions in Fig. 7.6.

As seen in the figure, while condition 1 clearly reaches a higher performance within the number of attempts allowed, condition 2 has a much higher initial rate of improvement. Condition 2 therefore also reaches a steady level of performance before condition 1. Both conditions therefore may be of interest, and it is hard to decisively say that one is preferred over the other. Such issues should be resolved before attempting full evaluations of this type of system, and this is

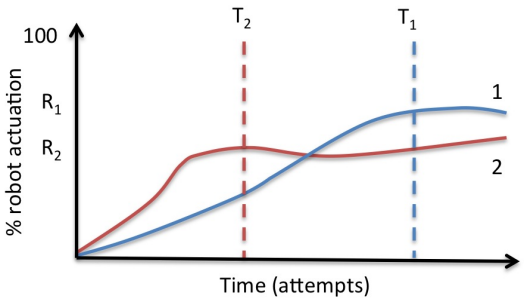


Fig. 7.6: *Hypothetical comparison of percentage actuation between two experiment conditions.*

one of the objectives of the ongoing work.

7.3 Using Sensorimotor Primitives to Drive Actuation?

7.3.1 Introduction

The behaviors learned in the previous chapter directly mapped recent sensory context to motor commands. That is, the overall behavior depended on the motor commands being exact enough to generate a recognisable sensory context for the next time step. Even small disturbances can in many cases therefore lead the system to sensory contexts where little or ambiguous information is available to drive the actuation further. In this section recent work on including sensorimotor primitives into the system will be outlined. Here the recent sensory context will be associated with simple sensor-driven behaviors. For example a "centering" behavior in-between obstacles. The activation of these sensorimotor primitives with the learned weights can then be used to influence the robot motor commands. This might be one way to learn a more robust actuation.

7.3.2 From Motor to Sensorimotor Primitives?

The idea that complex actions and movements are made up of combinations of simpler primitives is well established across several fields. In human (and animal) motor control, muscle synergies have been extensively studied. These are discrete groups of muscles that are activated as one unit, typically on the spinal level. It is then hypothesised that higher levels in the nervous system activate these muscle synergies, and that the linear combination of many such primitives can generate more complex movements and postures (Bizzi & Cheung 2013). This may help humans (and animals) orchestrate the thousands of motor units involved in complex movements, by reducing the number of variables that need to be controlled. It is therefore thought to help resolve the "degrees of freedom problem" (Bernstein 1967). Motor primitives can be found both at the behavioral, muscular and neural level, see (Flash & Hochner 2005) for an overview.

As mentioned in Chapter 6, motor primitives (also known as motor schemas, movement primitives etc) are also increasingly being employed in robot control. For example using point attractors and limit-cycles to represent discrete and rhythmic primitives, among other discussed in (Schaal et al. 2003). However, it seems useful to explore how the real sensory input of a complex mechatronic or biological system can drive the motor primitives. That is, how true sensorimotor coordination can emerge, that uses only the relevant information from the stream of high-dimensional sensor information to generate (and adapt) the relevant behaviors for a given situation. I will here use the name "sensorimotor primitives" to describe such behaviors. There is recent work in this direction (Zhong et al. 2014). This section will outline one approach for

including sensorimotor primitives in the work developed for this thesis.

7.3.3 Proposed Neural Network Changes

The proposed changes to the NN layout can be seen in Fig. 7.7. The main idea is to associate the recent sensor input with the level of "similarity" of the output of (many) behaviors and the motor signal caused by the teacher's demonstration. That is, each behavior produces a desired motor command (here represented by $\dot{x}_i$), which is compared to the actual command to produce a signal representing the "similarity". Here the negative exponential of the RMSE is used. Thus the more similar the two commands, the higher the signal. For each behavior, the signal linearly drives the activation of one of the neurons in the bottom layer of the learning NN. This activation is then associated with the recent sensory input as before. That is, the behaviors that most closely predict the actual motor commands are more strongly connected with the current sensory context.

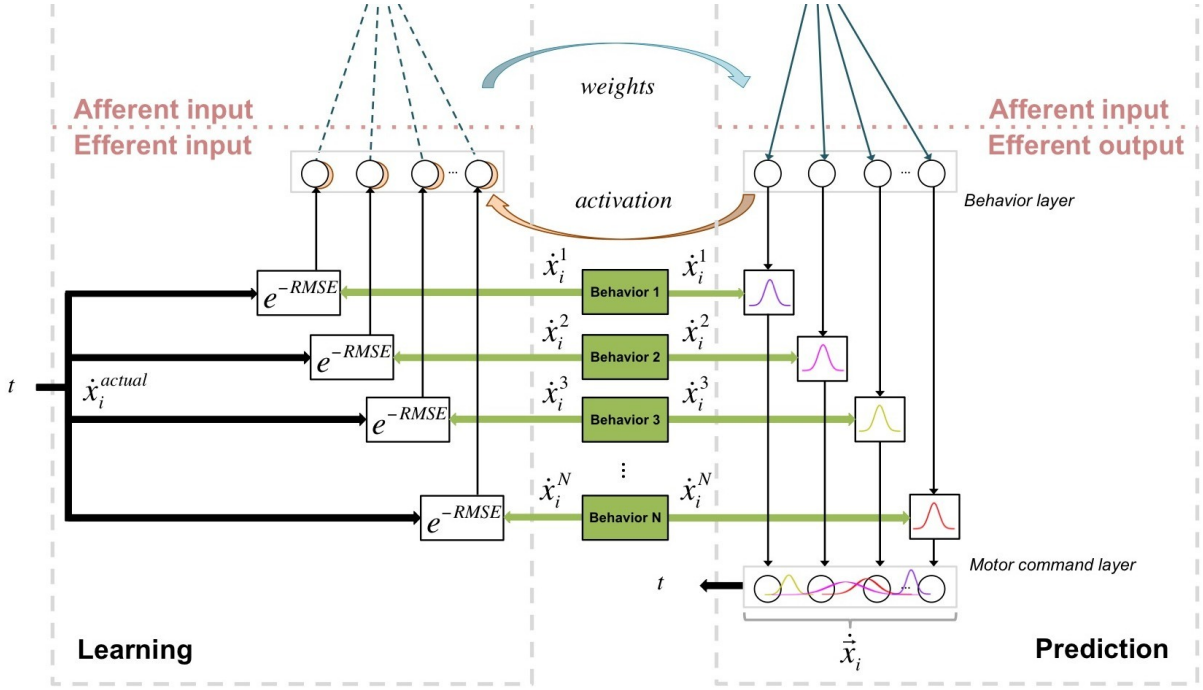


Fig. 7.7: *Preliminary idea for including behaviors representing sensorimotor primitives in the Neural Networks (NN). Layers for past and present sensory input not shown for clarity, and only one motor command ($\dot{x}_i$ with neurons $\dot{\vec{x}}_i$) included. Post superscripts are behavior identifiers.*

In the prediction NN, the activation from a given sensory context will be propagated over the learned weights to the neurons representing the different behaviors. The output of the bottom layer (the behavior layer) is now used to activate the different behaviors. That is, the output of each behavior (each of which are directly driven by sensory stimuli - so sensorimotor primitives) is represented as the mean of a Gaussian curve (for example). The height of the corresponding curve is driven by the activation in the corresponding neuron. The different Gaussian curves are then added up in a new motor command layer, in this way taking into account the contributions of all the behaviors. It is hoped that this type of architecture (or something similar) can make the actuation more robust and goal-oriented, by learning what combination of sensorimotor primitives (and their relative strength) should be activated in a given situation. I believe it can

also be combined with other layers directly mapping sensory context to motor commands, such as those presented in the previous chapters.

7.3.4 Towards an Experimental Setup

Several complicating factors arise when designing an experimental setup for evaluating a robot with an embedded NN architecture like the one described above. First, that the implementation should be on a physical robot, to be able to test the system with behaviors that use realistic sensory input. Using visual servoing to perform a "centering" behavior on a given color is trivial in a virtual environment, but is much more interesting in an environment with variable shades, variable light intensity and hue, and a less than perfect control of the robot end-effector. Second, to be able to test multiple heterogeneous behaviors, the tasks performed should be rich in sensory information and diverse. It seems to therefore make sense to move to tasks that are of a complexity similar to simple ADL. Such as picking up objects from different locations in a real kitchen. Fig. 7.8 shows four potential tasks of this type with the AMOR manipulator in the UC3M kitchen testbed.

(a) *Centering between obstacles.*

(b) *Slowing down before obstacle.*

(c) *Centering on a given color.*

(d) *Smoothing trajectory.*

Fig. 7.8: *Four example sensorimotor primitive applications.*

The task in Fig. 7.8a is intended to test centering behaviors between obstacles, here moving to grasp an object deep in a shelf. Many animal and human movements exhibit this property when moving in cluttered environments. The second task in Fig. 7.8b requires the system to slow down abruptly when close to the target to avoid hitting the wall behind. This is thus a representation of a Fitts' law (Fitts 1954) type task, where the speed-accuracy trade-off can be used as inspiration in the behavior design. Fig. 7.8c is another version of a "centering" type behavior, here on a given color. That is, the robot should learn to servo on bluish colors when picking up objects in similar situations (if this is a repeating trend). Finally, Fig. 7.8d is focused on carrying objects that need a smooth trajectory, such as a cup of coffee. Here for example the

well-known Minimum Jerk Model (Flash & Hogan 1985) could serve as inspiration. It is worth noting that multiple instances of each behavior can be used simultaneously (each with a distinct neuron), for example representing different speeds at which the behavior should be performed.

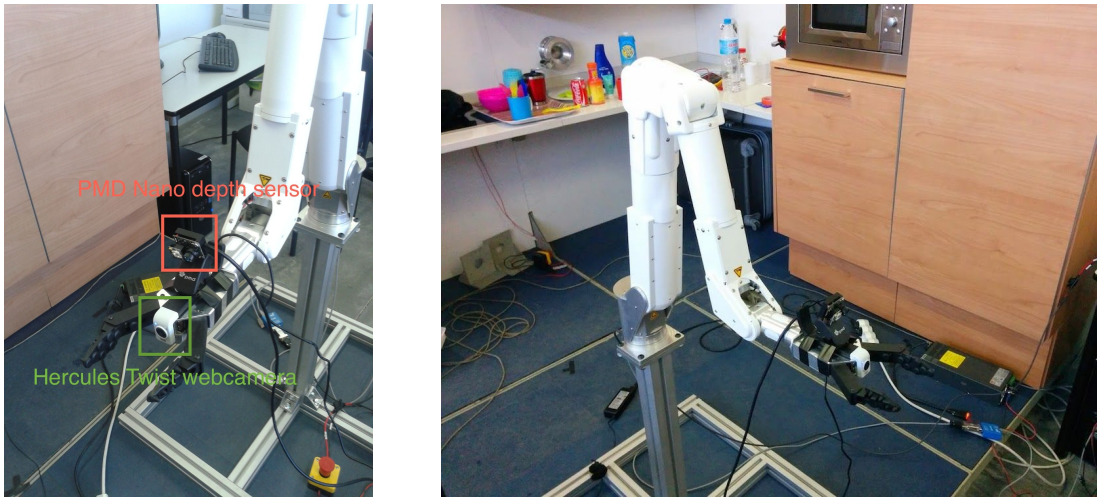
7.4 First Physical Implementation on an Assistive Manipulator

7.4.1 Introduction

This section will outline the current progress on implementing the concepts developed on a real assistive manipulator, the AMOR robot of Exact Dynamics in Holland. This work was unfortunately begun late in the thesis period due to a low amount of funding available. It is hoped that the work outlined can lead to practical systems that can adapt to, and assist, the user of similar physical robots in the near future.

7.4.2 Hand-Mounted Sensing

An assistive manipulator will in many situations have to rely on sensors that are built into the robot structure itself, for example if being used outside of a user's home. This distinguishes the application from many other in robotics, where one can reliably assume the presence, and exact relative location, of sensors in the environment. While external sensors can also be used here, it is interesting to see what can be done with onboard sensing. Two of the first sensors considered were a color camera and a Time Of Flight (TOF) depth sensor mounted on the hand of the robot. See Fig. 7.9.



(a) *Detail.* ***(b)*** *Overview.*

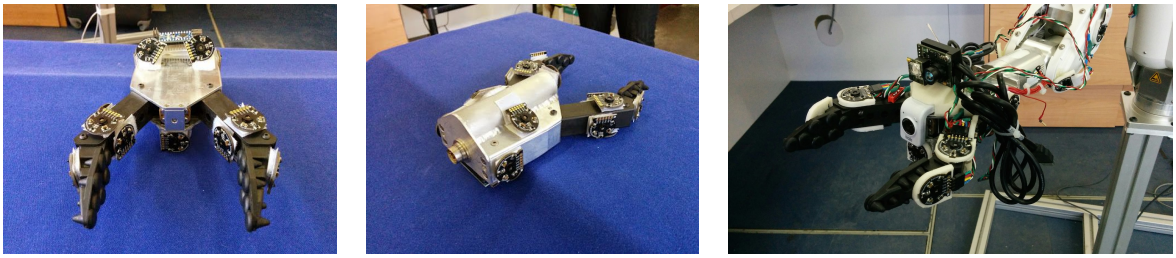
Fig. 7.9: *Two hand-mounted sensors considered for the AMOR robot manipulator, a Hercules Twist web camera and a PMD Nano depth sensor.*

The camera is a Hercules Twist, which has a form factor that allows it to be mounted between the fingers of the AMOR hand without interfering with the opening and closing of the fingers. It has a resolution of 1280 by 760 pixels, a wide-angle view, and an update rate of 30 Hz. The second sensor is a PMD Nano depth sensor, which has a low resolution (160 by 120 pixels), but a high frame rate (90 Hz) and a wide field of view (90° by 68 °). It is also very small (37 by 30 by 25 mm), allowing it to be mounted on the hand without significantly hindering the execution of tasks. The resolution is sufficient for close detection of obstacles and objects, and the sensor is capable of measuring distances from about 5-10 cm. Other sensors like the Microsoft Kinect

typically only work from 40 cm outwards. The sensor does not work well on black, transparent or shiny surfaces, like most distance sensing based on infrared light.

7.4.3 Distributed Proximity Sensing

Distributed proximity sensing is also being implemented on the AMOR, to test the SDAC concept on a physical assistive manipulator. A first implementation on the hand can be seen in Fig. 7.10, based on infrared Silicon Labs Si1143 sensors. Each sensor has its own digital circuit, and the sensors readings can be accessed over an I2C bus. A great advantage of these sensors is that they work from approx. 3-5 cm to 40 cm. As with the PMD Nano, they are noisy when used on black, transparent or shiny surfaces. A future implementation should therefore also include sensors based on other physical modalities. For example ultrasonic sensors.



(a) *Hand, front view.* ***(b)*** *Hand, bottom view.* ***(c)*** *First integration with all sensors.*

Fig. 7.10: *First implementation of distributed proximity sensing on the AMOR hand, based on infrared sensors (Silicon Labs Si1143).*

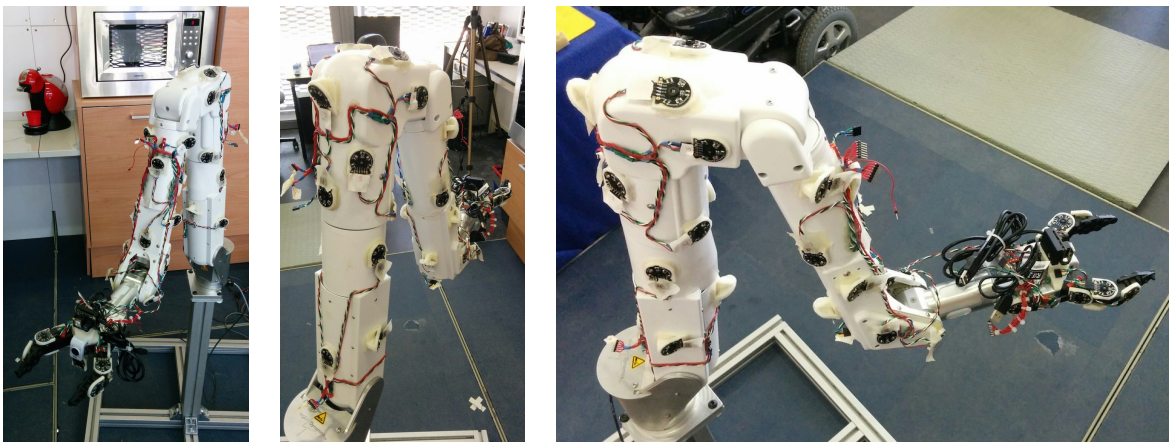
Fig. 7.11 shows the same type of proximity sensors on the body of the AMOR, with integration well underway. The current plan is to have local information-gathering on each link of the robot using Arduino Nano boards. Each board will communicate the readings to a central controller over wires or Bluetooth. Initial compatibility trials with the PMD Nano and the Microsoft Kinect is also under way. All three use infrared light around 850 nm. However, the current results indicate that there is little or no interference of the different sensors. More testing is needed to confirm this during usage on the robot platform. If true, there seems to be a great potential for combining static, hand-mounted and distributed infrared sensing, possibly also with hypersonic sensors. This would provide redundant and robust environmental sensing for assistive behaviors in this type of robot.

7.5 General Discussion

This chapter has presented some of the recent progress related to this thesis. While there is still much work before the approaches developed can be ran through clinical testing with end-users, there has at least been some progress made towards this goal. This section will discuss how to combine the ideas presented into a complete system, and the relationship with some of the related ideas in the literature.

7.5.1 Towards a Three-Level Adaptive Architecture

This thesis has included two approaches for adapting online to the user, one for helping to limit collisions, and one for attempting to predict and assist on the completion of tasks. Both behaviors seem useful, and it might be worth integrating both into a final system. Given that the user will always be in the control loop, the system would then have three levels of adaptation.



(a) Forearm. *(b)* "Lower arm". *(c)* Overview.

Fig. 7.11: *Current implementation of proximity sensors on the body of the AMOR manipulator.*

Making the analogy with our own nervous system, the levels could (in a loose way) be denoted as:

1. *"Conditioned Reflex":* The Shared Distributed Adaptive Control (SDAC) of Chapters 4, 5, and 7.
2. *"Muscle Memory":* The time-delayed NN with online Hebbian association of Chapters 6 and 7.
3. *Goal-Oriented Behavior:* The operator (typically the end-user).

One possible way of implementing this would be to include the SDAC approach as one (or several) of the behaviors in Fig. 7.7. That is, the Cartesian velocities allowed by the collision-limitation would be associated with the current sensory context. This context would likely include both visual and proprioceptive modalities, beyond the proximity sensing used by the behavior itself. The behavior would thus be activated based on a richer context, and also when the robot is in control of the movements. Unfortunately, it is not given that the adaptation of the three levels would result in useful behavior. However, I think this type of three-level architecture should be explored further.

7.5.2 Actuation Driving Sensing

One of the key properties of an embodied system, for example most animals, is that the actuation of the agent to a large degree drives and shapes the sensory feedback. This can simplify learning and control, by self-structuring the information received (Pfeifer et al. 2013). It is also a hot topic in psychology and neuroscience, and has been hypothesised as playing an important role in generating the "feel" of our everyday interactions with the world (O'Regan & Noë 2001). It is not a new idea (Dewey 1896), but seems to still be under-utilised in the design of robots. Looking at the work presented here, there are several positive effects of exploiting actuation to drive sensing.

In SDAC the distributed approach to sensing greatly simplifies the system. In fact, it seems to be the co-occurence of stimulation in the collision and proximity sensors that enables the use of a simple local Hebbian learning. The need for co-occurence also means that closely located sensors tend to connect the strongest. Taking advantage of such concurrency in time and space,

the learning can directly single out causal links between proximity and collisions, that again can help drive meaningful behaviour. Establishing such causal links is a more complex task when a more static and centralised approach to sensing is attempted. As an example, there is typically no causal link between the value of a given pixel (color or depth) in a static sensor and a collision between a given part of the robot and the environment.

Also in the approach for interactive learning of tasks proposed there are clear examples of actuation driving sensing. There is for example no explicit concept of a stored time-dictated trajectory. The actuation at each step in time is driven by the recent sensorial input (can also be an efferent copy), which again depends on the previous actuation. The approach was also implemented and tested on an embodied platform, the iCub, where for example the head movements (driven by simple reflex behaviours or higher-order commands) drives the visual information received.

7.5.3 The Potential Role of Mutual Adaptation

Throughout the thesis an emphasis has been placed on keeping all changes in the robot behavior occur during the user interaction. This was also included as a key design driver, i.e. enabling online incremental adaptation in the robot. One of the main motivations is to keep the system predictable from the user's perspective, but also to provide good feedback on the state and changing intent of the robot, a second key design driver. Given that the user is also able to adapt, we then have a system with the potential for mutual adaptation between the user and the robot. This was clearly the case with SDAC, where both the system adapted (by increasing the use of proximity sensors after collisions) and the operator adapted (by for example moving faster when receiving appropriate assistance). The result was improved performance in most cases, indicating that mutual adaptation can have a practical use in this application.

Whether similar or better improvements may be possible with a system that learns offline I do not know, but I have a feeling it will at least be less intuitive for the user. I think the concept of mutual adaptation is especially relevant to the closely related field of prosthetics. Here the user and the limb has to find a "common language", often through nerve endings that have been used for very different purposes previously. Mutual adaptation allows both the encoder and decoder of such a communications channel to be tuned, likely making the probability of finding a good protocol higher. However, mutual adaptation requires good feedback to the user on current performance (Yokoi et al. 2004). I believe this can be achieved in an assistive manipulator through the type of visual, auditory and haptic feedback explored here.

8. CONCLUSIONS

8.1 Overview

The central theme of this thesis is to enable adaptation in physically assistive robot manipulators. It has been shown that the physical structure of the symmetric ASIBOT manipulator can be optimised for specific tasks and obstacles. This type of task-oriented design could perhaps lead to manipulators that are better adapted to the user's needs, as compared to a less quantitative design approach. The second type of adaptation explored was an aid for limiting collisions, adapted to each robot operator. This behavior was inspired by the speed-accuracy trade-off of human movements, in effect allowing the operator to move faster by helping limit collisions similar to those previously experiences. The work has also showed that one of the most well-known implications of the speed-accuracy trade-off, Fitts' law, seems to also apply to combined rotational and translational movements. Finally, the thesis outlined an approach for robot task learning in real-time with a human teacher. The approach associates high-dimensional sensor input with robot actuation while the task is being demonstrated, leading to more interactive learning. It may also be useful for recognising and assisting on tasks made by the user of an assistive manipulator, as outlined in the later chapters. The author has also developed and applied different approaches for modelling and benchmarking assistive manipulators, in an attempt to move towards replicable experiments. It is possible and expected that future experiments with real assistive manipulators and disabled persons will highlight new aspects about the work presented. Nevertheless, the simulation studies performed are necessary prerequisites to clinical trials, and have already helped the development in this direction. I believe that making these robots safer and more effective through adaptation could increase the independence of the end-users in their daily lives.

8.2 Contributions

The main contributions of this work are:

1. *Shared Distributed Adaptive Control (SDAC) of assistive manipulators.* Perhaps the main contribution of this development was to show that the DAC approach has potential also in a practical robot application. The mutual adaptation occurring between the user and the system is also interesting, with a potential for rapidly finding a "good" collision limitation behaviour for each user.

2. *Approaches for interactive and context-driven learning for high-DOF robot platforms.* This development was focused on attempting to learn what the user would like to do, and to assist in the completion of tasks. The contribution was perhaps mainly in the ability to learn from a high-dimensional sensory context in real-time on a complex humanoid robot platform. The tasks remain simple, but I nonetheless believe similar approaches may be of interest for assistive manipulators (and other platforms) in the near future.

3. *First demonstration of Fitts' Law for combined translational and rotational movements.* While the law has been applied to a range of movements involving either rotations or

translations, this study showed it has some power of explanation also for combined movements. This might also help our understanding of more complex movements in the future.

4. *Promoting replicable experiments in robotics with system models, metrics and protocols.* In all 7 controlled human experiments were included in this thesis (more were conducted), with 5-13 participants in each. I hope these experiments, together with the performance metrics and system models developed, can help shed some light on this type of closely coupled human-machine systems. An open repository for sharing similar experiments has been set up at: http://throughput.sourceforge.net.

5. *Extending grid-based task-oriented kinematic optimization to symmetric climbing robots.* This work extended the efficient grid-based approach for kinematic optimisation to symmetric robots, enabling a task-oriented design of an ASIBOT-type climbing manipulator. This type of design process can be used to better adapt such robots to typical tasks end environments.

8.3 Suggested Future Lines of Research

The future lines of research enabled by this work include:

1. *Adaptation and shared control with end-users of assistive manipulators.* It is not given that the results obtained will be similar for real end-users, but I believe the thorough testing with different simulated disabilities and time-delays shows that there is a potential. Funding is needed to perform extensive testing with the physical system, preferably over a long time-frame to enable the user to adapt to (and with) the system. Other applications like exoskeletons and prostheses should also be explored.

2. *The role of embodiment and context in animal and robot learning.* An interesting future line of research is to explore how the robot body (embodiment) shapes the information being received by the robot through its sensory apparatus. And particularly how robot learning can be simplified by using sensors that are distributed over, and highly coupled to, the embodiment.

3. *"Shared" motion planning.* Autonomous manipulation of objects in real-world environments remains a difficult challenge. One approach forward could be to include the user in the motion planning process, in a similar way to the lower-level shared control used here. For example by allowing the user to adjust the planned trajectory of the robot during execution, with immediate feedback on the changes to the trajectory (e.g. visual).

4. *Information theoretic modelling of high-DOF movement task complexity.* The original Information Theoretic interpretation of Fitts' law is not commonly used, but an information-based view of complex human (and robot) movements might still be useful. Along the lines of information-based views of task complexity of 2D scenes (Lampe & Chatila 2006), of control systems (Touchette & Lloyd 2004), of networked embodied intelligence (Bonsignorio 2007), and of the self-structuring of embodied cognitive networks (Bonsignorio 2013).

5. *The role of movement direction on combined task performance.* The Fitts' law experiment setup used here included rotations and translations in only one direction. The effect of different combinations of directions seems like an interesting research topic. There is already a follow-up work in this direction (Nguyen & Kipp 2014). Another possible topic is the role of parallel versus sequential execution of the movement components, and what factors can trigger a switch between such behaviours.

6. *Task-oriented optimization of docking station placement.* Given that ASIBOT is able to climb between docking stations, the placement of these stations should likely be taken explicitly into account in a task-oriented design process. This may also drive the kinematic design of the robot.

8.4 Relevant Publications

8.4.1 International Journals

1. M.F. Stoelen, V.F. Tejada, A. Jardón, C. Balaguer, and F. Bonsignorio, "An Approach for Modeling and Benchmarking Shared Control of Assistive Manipulators," under review, *IEEE Transactions on Robotics*.

2. A. Jardón, M.F. Stoelen, F. Bonsignorio, and C. Balaguer, "Task-Oriented Kinematic Design of a Symmetric Assistive Climbing Robot," short paper, *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1132-1137, 2011.

3. M.F. Stoelen and D.L. Akin, "Assessment of Fitts' Law for Quantifying Combined Rotational and Translational Movements," *Human Factors, The Journal of the Human Factors and Ergonomics Society*, vol. 52, no. 1, pp. 63-77, 2010.

8.4.2 International Peer-Reviewed Conferences

1. M.F. Stoelen, V.F. Tejada, A. Jardón, F. Bonsignorio and C. Balaguer, "Distributed Sensing, Learning and Control in an Assistive Manipulator," *IEEE Multi-Conference on Systems and Control*, Antibes, France, 2014.

2. M.F. Stoelen, D. Marocco, A. Cangelosi, F. Bonsignorio and C. Balaguer, "Predictive Hebbian Association of Time-Delayed Inputs with Actions in a Developmental Robot Platform," *IEEE International Joint Conference on Neural Networks (IJCNN)*, Beijing, China, 2014.

3. M.F. Stoelen, V.F. Tejada, A. Jardón, F. Bonsignorio and C. Balaguer, "Adaptive Collision-Limitation Behavior for an Assistive Manipulator," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1143-1148, Tokyo, Japan, 2013.

4. M.F. Stoelen, F. Bonsignorio, C. Balaguer, D. Marocco and A. Cangelosi, "Online Learning of Sensorimotor Interactions using a Neural Network with Time-Delayed Inputs," *IEEE Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, San Diego, USA, 2012.

5. M.F. Stoelen, V.F. Tejada, A. Jardón, F. Bonsignorio and C. Balaguer, "Benchmarking Shared Control for Assistive Manipulators: From Controllability to the Speed-Accuracy Trade-Off," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4386-4391, Vilamoura, Portugal, 2012.

6. M.F. Stoelen, B. Dillow, S.E. Jacobs and D.L. Akin, "Interface for EVA Human-Machine Interaction," *38th International Conference on Environmental Systems (ICES)*, San Francisco, USA, 2008.

8.4.3 Workshops

1. M.F. Stoelen, D. Marocco, A. Cangelosi, F. Bonsignorio and C. Balaguer, "Hebb-like Learning for the Grounding of High-Level Symbols in Sensorimotor Trajectories," *Proceedings of the Post-Graduate Conference on Robotics and Development of Cognition*, Lausanne, Switzerland, 2012.

2. M.F. Stoelen, A. Jardón, V. Fernández, C. Balaguer and F. Bonsignorio, "An information-theoretic approach to modeling and quantifying assistive robotics HRI," *late breaking report, Proceedings of the 6th international conference on Human-robot interaction (HRI)*, pp. 257-258, Lausanne, Switzerland, 2011.

3. M.F. Stoelen, A. Jardón, J.G. Victores, C. Balaguer and F. Bonsignorio, "Information Metrics for Assistive Human-In-The-Loop Cognitive Systems," *Workshop on Good Experimental Methodology in Robotics and Replicable Robotics Research, Robotics Science and Systems (RSS)*, Zaragoza, Spain, 2010.

4. M.F. Stoelen, A. Jardón, F. Bonsignorio, J.G. Victores, C. Monje and C. Balaguer, "Towards an Enabling Multimodal Interface for an Assistive Robot," *Workshop on Multimodal Human-Robot Interfaces, IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, Alaska, 2010.

APPENDICES

A. METHOD FOR FITTS' LAW STUDY

A.1 Task Parameters Used

Tab. A.1: *All combinations of distances (A in cm and α in degrees) and tolerances (W in cm and ω in degrees) used for experiments 1, 2, and 3.*

Combination	Experiment 1			Experiment 2			Experiment 3					
	A	W	ID_t	α	ω	ID_r	A	W	ID_t	α	ω	ID_r
1	4.8	0.6	3.1	40	3	3.8	12.7	0.8	4.1	50	4	3.8
2	4.8	1.3	2.2	40	6	2.9	12.7	1.6	3.2	50	4	3.8
3	4.8	1.9	1.8	40	9	2.4	4.8	0.8	2.8	50	4	3.8
4	4.8	2.5	1.5	40	12	2.1	4.8	1.6	2.0	50	4	3.8
5	9.5	0.6	4.0	80	3	4.8	12.7	0.8	4.1	50	12	2.4
6	9.5	1.3	3.1	80	6	3.8	12.7	1.6	3.2	50	12	2.4
7	9.5	1.9	2.6	80	9	3.3	4.8	0.8	2.8	50	12	2.4
8	9.5	2.5	2.2	80	12	2.9	4.8	1.6	2.0	50	12	2.4
9	14.3	0.6	4.6	120	3	5.4	12.7	0.8	4.1	130	4	5.1
10	14.3	1.3	3.6	120	6	4.4	12.7	1.6	3.2	130	4	5.1
11	14.3	1.9	3.1	120	9	3.8	4.8	0.8	2.8	130	4	5.1
12	14.3	2.5	2.7	120	12	3.5	4.8	1.6	2.0	130	4	5.1
13	19.1	0.6	5.0	160	3	5.8	12.7	0.8	4.1	130	12	3.6
14	19.1	1.3	4.0	160	6	4.8	12.7	1.6	3.2	130	12	3.6
15	19.1	1.9	3.5	160	9	4.2	4.8	0.8	2.8	130	12	3.6
16	19.1	2.5	3.1	160	12	3.8	4.8	1.6	2.0	130	12	3.6
Minimum	4.8	0.6	1.5	40	3	2.1	4.8	0.8	2.0	50	4	2.4
Maximum	19.1	2.5	5.0	160	12	5.8	12.7	1.6	4.1	130	12	5.1

A.2 Experiment 1: Translational Movements

Participants The participants were 12 students and staff of the University of Maryland, 6 male and 6 female. None of the participants were included in the other two experiments in this study. All participants were right-handed and were aged between 18 and 35 with a mean of 22.3 years. There were 9 participants with corrected vision and 8 had previous experience with 3D input devices. All gave their informed consent to participate according to the regulations of the

University of Maryland's Institutional Review Board (IRB). Participants not employed by the Space Systems Laboratory were paid $10 for their participation.

Apparatus The experiment was conducted on an Apple iMac workstation in an office environment. The participants worked on a 20-inch external liquid crystal display monitor (Dell 2007WFP) with a 60 Hz refresh rate. The input device was a magnetic Ascension Flock of Birds sensor system sampling at a rate of 70 Hz. The system has a specified translational and rotational accuracy of 1.8 mm (0.8 mm resolution) and 0.5° (0.1° resolution) respectively. The sensor of this system was held in the participant's dominant hand and measured one translational DOF, the left to right position relative to the display in front of the participant. These measurements were used to update the left to right position of a cursor on the display. A calibration was performed to ensure that the apparent displacement of the cursor on the screen corresponded with that of the hand-held sensor. The participants also manipulated a button updating at 100 Hz with the alternate hand. See Figure A.1 for the physical setup of the experiment. Computer generated voice feedback was used to inform the participant about the successful or unsuccessful completion of each trial.

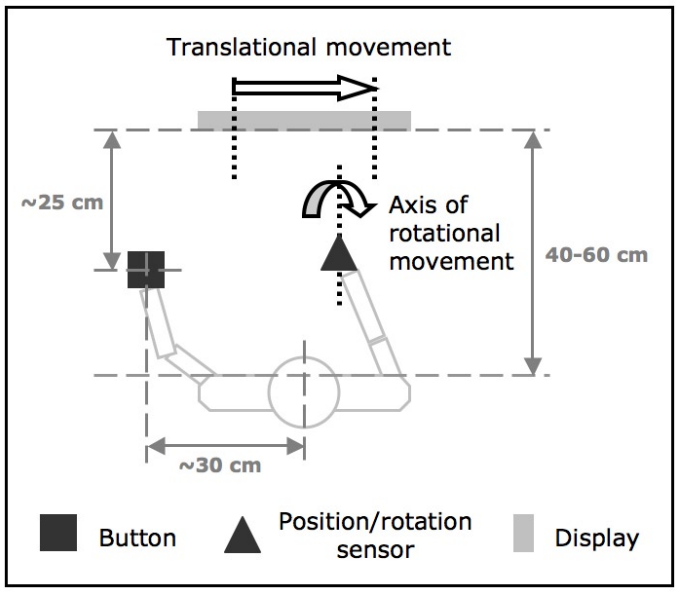


Fig. A.1: *Overhead view of general experiment setup in right-handed configuration.*

Stimuli The stimuli presented on the display can be seen in Figure 2.6a. A cursor in the form of a small disc on the display indicated the left to right position of the hand-held sensor held by the participant, as described above. Two vertical lines on the display were used to indicate the area from which each trial should start and two vertical lines were used to indicate the area within which each trial should finish. The distance between the center of the two areas (movement distance) and the size of the areas (movement tolerance) were varied randomly.

Procedure The participants were instructed to start each trial with the disc within the start area on the display. When ready the participant would then press and hold a button with the alternate hand, indicating the start of the trial. If the button was pressed while the disc was within the start area a timer was started. If the disc was outside the start area when the button was pressed, the participant was notified and the trial restarted. The successful start of the

timer was indicated by a change in color of all objects on the display from blue to green. The participants were told that the goal of each trial was to then move the disc to within the finish area and let go of the button. If the button was released outside the finish area, the trial was restarted, the participant notified and an error recorded. If the button was released inside the finish area, the timer was stopped and the participant notified of the successful completion of the trial. The movements were from left to right. The participants were instructed to keep the hand holding the sensor (the dominant hand) off the table during movements to prevent obstruction. The participants were instructed to emphasize accuracy over speed (consistent with instructions given in (Fitts 1954)) to achieve an error level of around 2 percent. Feedback on the actual error percentage was given during the initial training session to help the participants adjust their performance.

Each participant's session lasted about one hour, starting with a thorough brief of the task procedures and a short questionnaire about basic personal information. After completing the task, each participant filled out a questionnaire about the experiment, and was also asked to provide suggestions for improving the experiment. Each experiment included 16 different combinations of distances and tolerances, each repeated 12 times for non-training trials. All participants were first given 48 trials for training, to get acquainted with the task and the experimental setup. The participants then performed 96 trials, followed by a five-minute break and 96 more trials. The participants were also allowed to take short breaks in-between each trial as needed.

A.3 Experiment 2: Rotational Movements

Participants The participants were 13 students and staff of the University of Maryland, 5 male and 8 female. None of the participants were included in the other two experiments in this study. 12 participants were right-handed and one left-handed. All were aged between 19 and 48, with a mean of 23.5 years. There were 12 participants with corrected vision and 6 had previous experience with 3D input devices.

Apparatus The participants worked with the same physical apparatus as used in experiment 1. For this experiment the hand-held sensor measured one rotational DOF only, the roll axis of the hand-held sensor from the participant's point of view, see Figure A.1. Thus, the task required rotary movements mainly about the longitudinal axis of the forearm, similar to turning a doorknob or using a standard screwdriver. No physical restriction was placed on the movement, allowing the participants to use both fingers and wrist to produce the rotation. The measurements from the sensor were used to update the angle of a cursor on the display. The apparent angle of the cursor on the screen corresponded with that of the hand-held sensor.

Stimuli The stimuli presented on the display can be seen in Figure 2.6b. A cursor in the form of a line on the display indicated the angle of the hand-held sensor held by the participant, as described above. Two radial lines on the display were used to indicate the area from which each trial should start, and two radial lines were used to indicate the area within which each trial should finish. The angle between the center of the two areas (movement distance) and the angle between the two lines representing an area (movement tolerance) were varied randomly.

Procedure The participants followed the same procedure as in experiment 1, except for the type of movement performed. The rotational movements were clockwise for right-handed participants. Left-handed participants used a mirror-image setup, performing the same forearm/finger supination. The participants were instructed to keep the hand holding the sensor (the dominant hand) off the table during movements to prevent obstruction.

A.4 Experiment 3: Combined Movements

Participants The participants were 13 students and staff of the University of Maryland, 11 male and 2 female. None of the participants were included in the other two experiments in this study. 12 participants were right-handed and one left-handed. All were aged between 18 and 32, with a mean of 22.2 years. There were 12 participants with corrected vision and 9 had previous experience with 3D input devices.

Apparatus The participants worked with the same physical apparatus as used in experiment 1 and 2. For this experiment the hand-held sensor measured 2 DOF only, the translational DOF used in experiment 1 and the rotational DOF used in experiment 2. Thus the sensor measured the left to right position relative to the display in front of the participant as well as the roll axis of the hand-held sensor. These measurements were used to update the left to right position of a translational cursor as well as the angle of a rotational cursor on the display. As for experiment 1 and 2, the apparent position and rotation of the respective cursors corresponded with that of the hand-held sensor.

Stimuli The stimuli presented on the display can be seen in Figure 2.6c. A cursor in the form of a disc indicated the left to right position, and a cursor in the form of a line indicated the angle of the hand-held sensor held by the participant, as described above. The rotational axis of the rotational cursor was kept centered at the position of the translational cursor at all times. Thus the rotational tasks, including the radial lines representing the rotational start and finish areas, moved with the translational cursor. This was done to reduce the eye movements needed to coordinate the two tasks, and to provide a visualization of the combined movement that was as similar as possible to the actual movements of the hand-held sensor. The movement distances and tolerances were varied randomly.

Procedure The participants followed the same procedure as in experiment 1 and 2 except for the type of movement performed. The translational movements were from left to right and the rotational movements clockwise for right-handed participants. Left-handed participants used a mirror-image setup, performing the same arm abduction and forearm/finger supination. The participants were instructed to keep the hand holding the sensor (the dominant hand) off the table during movements to prevent obstruction.

B. LIST OF COMMON ACRONYMS

ADL Activities of Daily Living. 1, 32–34, 127

AI Artificial Intelligence. 7, 8

DAC Distributed Adaptive Control. 57, 58, 76, 93, 133

DH Denavit-Hartenberg. 38, 40, 42–44, 49

DOF Degrees Of Freedom. 1, 2, 5, 7, 17, 19, 20, 22, 23, 29–31, 38, 40, 44–46, 48–55, 62, 71, 77, 81, 84, 91, 99, 102, 109, 133, 134, 140–142

ID Index of Difficulty. 19, 23–26, 29

MT Mean Time. 19, 29, 30, 63, 66, 72, 84, 85, 88

NN Neural Network. 7, 9, 57, 58, 60, 68, 71, 80, 81, 84, 87, 96–100, 103, 105–108, 110, 113–119, 122, 126, 127, 130

RMSE Root Mean Square Error. 101, 109, 122, 126

SDAC Shared Distributed Adaptive Control. 93, 121, 129–131

TOD Task-Oriented Design. 37, 38, 40, 52

UC3M Universidad Carlos III de Madrid. 45, 61, 69, 82, 86, 127

REFERENCES

Abbink, D. A., Mulder, M. & Boer, E. R. (2012), 'Haptic shared control: smoothly shifting control authority?', *Cognition, Technology & Work* **14**(1), 19–28.

Accot, J. & Zhai, S. (1997), 'Beyond Fitts' law: models for trajectory-based HCI tasks', *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* **97**, 295–302.

Akgün, B., Tunaoglu, D. & Sahin, E. (2010), 'Action recognition through an action generation mechanism', *International Conference on Epigenetic Robotics (EPIROB)* .

Akin, D., Roberts, B., Pilotte, K. & Baker, M. (2003), 'Robotic Augmentation of EVA for Hubble Space Telescope Servicing', *AIAA Space 2003 Conference & Exposition* .

Amigoni, F., Reggiani, M. & Schiaffonati, V. (2009), 'An insightful comparison between experiments in mobile robotics and in science', *Autonomous Robots* **27**(4), 313–325.

Asada, M., MacDorman, K., Ishiguro, H. & Kuniyoshi, Y. (2001), 'Cognitive developmental robotics as a new paradigm for the design of humanoid robots', *Robotics and Autonomous Systems* **37**(2), 185–193.

Asfour, T., Azad, P. & Vahrenkamp, N. (2008), 'Toward humanoid manipulation in human-centred environments', *Robotics and Autonomous Systems* **56**, 54–65.

Ashby, W. (1957), 'Pavlov Reconditioned', *The British Journal for the Philosophy of Science* **8**(31), 249–252.

Ashby, W. (1960), *Design for a brain*, 2nd edn, Wiley, New York.

Ay, N., Bertschinger, N., Der, R., Güttler, F. & Olbrich, E. (2008), 'Predictive information and explorative behavior of autonomous robots', *The European Physical Journal B* **63**(3), 329–339.

Balaguer, C., Gimenez, A., Jardon, A., Cabas, R. & Correal, R. (2005), Live experimentation of the service robot applications for elderly people care in home environments, *in* 'Proc. of the International Conference on Intelligent Robots and Systems', pp. 2345–2350.

Barreto, G., Araújo, A. & Kremer, S. (2003), 'A taxonomy for spatiotemporal connectionist networks revisited: the unsupervised case', *Neural Computation* **15**(6), 1255–1320.

Basdogan, C. & Srinivasan, M. A. (2002), 'Haptic rendering in virtual environments', *Handbook of virtual environments* pp. 117–134.

Beamish, D., Bhatti, S. A., MacKenzie, I. S. & Wu, J. (2006), 'Fifty years later: A neurodynamic explanation of Fitts' law', *Journal of the Royal Society, Interface / the Royal Society* **3**(10), 649–54.

Belpaeme, T. & Morse, A. (2010), 'Time will tell - why it is too early to worry', *Interaction Studies* **11**(2), 191–195.

Bennett, C. & Shor, P. (1998), 'Quantum information theory', *IEEE Transactions on Information Theory* **44**(6), 2724–2742.

Bernstein, N. (1967), *The Coordination and Regulation of Movements*, Pergamon Press, Oxford, UK.

Bialek, W., Nemenman, I. & Tishby, N. (2001), 'Predictability, complexity, and learning', *Neural computation* **13**(11), 2409–63.

Bien, Z., Chung, M.-J., Chang, P.-H., Kwon, D.-S., Kim, D.-J., Han, J.-S., Kim, J.-H., Kim, D.-H., Park, H.-S., Kang, S.-H., Lee, K. & Lim, S.-C. (2004), 'Integration of a rehabilitation robotic system (kares ii) with human-friendly man-machine interaction units', *Auton. Robots* **16**(2), 165–191.

Bizzi, E. & Cheung, V. C. K. (2013), 'The neural origin of muscle synergies', *Frontiers in computational neuroscience* **7**(April), 51.

Bonarini, A., Burgard, W., Fontana, G., Matteucci, M., Sorrenti, D. & Tardos, J. (2006), Rawseeds: Robotics advancement through web-publishing of sensorial and elaborated extensive data sets, *in* 'Proceedings of IROS'06 Workshop on Benchmarks in Robotics Research'.

Bonsignorio, F. (2007), Preliminary considerations for a quantitative theory of networked embodied intelligence, *in* '50 years of artificial intelligence', Springer Berlin Heidelberg, pp. 112–123.

Bonsignorio, F. (2013), 'Quantifying the evolutionary self-structuring of embodied cognitive networks', *Artificial life* **19**(2), 267–289.

Bonsignorio, F., Hallam, J. & del Pobil, A. (2009), Defining the requisites of a replicable robotics experiment, *in* 'RSS2009 Workshop on Good Experimental Methodologies in Robotics'.

Bowman, D. A., Kruijff, E., Laviola, J. J. & Poupyrev, I. (2004), *3D user interfaces: Theory and practice*, Addison Wesley.

Bruyninckx, H. (2001), Open robot control software: the orocos project, *in* 'Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on', Vol. 3, IEEE, pp. 2523–2528.

Burgsteiner, H. (2006), 'Imitation learning with spiking neural networks and real-world devices', *Engineering Applications of Artificial Intelligence* **19**(7), 741–752.

Calinon, S. (2009), *Robot Programming by Demonstration: A Probabilistic Approach*, EPFL/CRC Press, Lausanne, Switzerland.

Calinon, S. & Billard, A. (2007), 'Incremental learning of gestures by imitation in a humanoid robot', *Proceeding of the ACM/IEEE international conference on Human-robot interaction - HRI '07* pp. 255–262.

Card, S. K., English, W. K. & Burr, B. (1978), 'Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys, for text selection on a CRT', *Ergonomics* **21**(8), 601–613.

Carignan, C. R. & Krebs, H. I. (2006), 'Telerehabilitation robotics: Bright lights, big future?', *The Journal of Rehabilitation Research and Development* **43**(5), 695–710.

Carlson, T. & Demiris, Y. (2008), 'Human-wheelchair collaboration through prediction of intention and adaptive assistance', *2008 IEEE International Conference on Robotics and Automation* pp. 3926–3931.

Carlson, T. & Demiris, Y. (2012), 'Collaborative control for a robotic wheelchair: evaluation of performance, attention, and workload.', *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics* **42**(3), 876–88.

Chang, P. H. U. N. P. & Park, H. H.-s. (2003), 'Development of a Robotic Arm for Handicapped People : A Task-Oriented Design Approach', *Autonomous robots* **15**(1), 81–92.

Chaves, E., Koontz, A., Garber, S., Cooper, R. & Williams, A. (2003), Clinical evaluation of a wheelchair mounted robotic arm, *in* 'Technical report, Univ. of Pittsburgh'.

Chen, T. L., Ciocarlie, M. T., Cousins, S. B., Grice, P. M., Hawkins, K. P., Hsiao, K., Kemp, C. C., King, C.-H., Lazewatsky, D. A., Leeper, A., Nguyen, H., Paepcke, A., Pantofaru, C., Smart, W. D. & Takayama, L. (2013), 'Robots for humanity: Using assistive robotics to empower people with disabilities', *IEEE Robot. Automat. Mag.* **20**(1), 30–39.

Chen, T. L., Jain, A. & Kemp, C. C. (2010), 'Towards an assistive robot that autonomously performs bed baths for patient hygiene', *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 319–324.

Cheung, E. & Lumelsky, V. V. (1989), 'Proximity sensing in robot manipulator motion planning: system and implementation issues', *Robotics and Automation, IEEE Transactions on* **5**(6), 740–751.

Chocron, O. & Bidaud, P. (1997), Evolutionary algorithms in kinematic design of robotic systems, *in* 'Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on', Vol. 2, pp. 1111–1117.

Choi, Y. S., Anderson, C. D., Glass, J. D. & Kemp, C. C. (2008), Laser pointers and a touch screen: Intuitive interfaces for autonomous mobile manipulation for the motor impaired, *in* 'Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility', ACM, pp. 225–232.

Ciocarlie, M., Hsiao, K., Leeper, A. & Gossow, D. (2012), 'Mobile manipulation through an assistive home robot', *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 5313–5320.

Collins, S., Ruina, A., Tedrake, R. & Wisse, M. (2005), 'Efficient bipedal robots based on passive-dynamic walkers', *Science (New York, N.Y.)* **307**(5712), 1082–1085.

Commision, E. (2012), 'Public attitudes towards robots', *Special Eurobarometer 382* .

Cover, T. M. & Thomas, J. A. (2012), *Elements of information theory*, John Wiley & Sons.

Crossan, A. & Murray-Smith, R. (2004), Variability in wrist-tilt accelerometer based gesture interfaces, *in* 'Proceedings of Mobile Human-Computer Interaction-MobileHCI 2004: 6th International Symposium', pp. 144–155.

Crossman, E. & Goodeve, P. (1983), 'Feedback control of hand-movement and fitts' law', *Quarterly Journal of Experimental Psychology* **35**(2), 251–278.

Davis, J., Smyth, C. & McDowell, K. (2010), 'The Effects of Time Lag on Driving Performance and a Possible Mitigation', *IEEE Transactions on Robotics* **26**(3), 590–593.

Dellon, B. & Matsuoka, Y. (2007), 'Prosthetics, exoskeletons, and rehabilitation', *Robotics & Automation Magazine, IEEE* **14**(1), 30–34.

Demeester, E., Huntemann, A., Vanhooydonck, D., Vanacker, G., Brussel, H. V. & Nuttin, M. (2008), 'User-adapted plan recognition and useradapted shared control: A bayesian approach to semi-autonomous wheelchair driving', *Journal of Autonomous Robots* **24**(2), 193–211.

Dennerlein, J. T., Martin, D. B. & Hasser, C. (2000), 'Force-feedback improves performance for steering and combined steering-targeting tasks', *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '00* **2**(1), 423–429.

Dewey, J. (1896), 'The reflex arc concept in psychology', *Psychological review* **3**(4), 357.

Diankov, R. (2010), Automated Construction of Robotics Manipulation Programs, PhD thesis, Carnegie Mellon University. Robotics Institute.

Dodds, T. A., Martin, D. P., Stolov, W. C. & Deyo, R. A. (1992), 'A validation of the functional independence measurement and its performance among rehabilitation inpatients', *Stroke* **7**, 65–75.

Drury, C. G. (1975), 'Application of fitts' law to foot-pedal design', *Human Factors: The Journal of the Human Factors and Ergonomics Society* **17**(4), 368–373.

Everett, S. & Dubey, R. (1998), 'Human-Machine Cooperative Telerobotics Using Uncertain Sensor or Model Data', *Proceedings of the 1998 IEEE International Conference on Robotics & Automation* **2**(May), 1615–1622.

Faisal, A. A., Selen, L. P. J. & Wolpert, D. M. (2008), 'Noise in the nervous system', *Nat Rev Neurosci* **9**(4), 292–303.

Felton, E. a., Radwin, R. G., Wilson, J. a. & Williams, J. C. (2009), 'Evaluation of a modified Fitts law brain-computer interface target acquisition task in able and motor disabled individuals', *Journal of neural engineering* **6**(5), 56002.

Findlater, L. & McGrenere, J. (2004), 'A comparison of static, adaptive, and adaptable menus', *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04* **6**(1), 89–96.

Fitts, P. M. (1954), 'The information capacity of the human motor system in controlling the amplitude of movement', *J Exp Psychol* **47**(6), 381–391.

Flash, T. & Hochner, B. (2005), 'Motor primitives in vertebrates and invertebrates', *Current opinion in neurobiology* **15**(6), 660–666.

Flash, T. & Hogan, N. (1985), 'The coordination of arm movements: An experimentally confirmed mathematical model', *The Journal of Neuroscience* **5**(7), 1688–1703.

Fong, T. & Nourbakhsh, I. (2005), 'Interaction challenges in human-robot space exploration', *Interactions* **12**(2), 42–45.

Gallego, J., Rocon, E., Belda-Lois, J. M. & Pons, J. L. (2013), 'A neuroprosthesis for tremor management through the control of muscle co-contraction', *Journal of NeuroEngineering and Rehabilitation* **10**(1), 36.

Gan, K.-C. C. & Hoffmann, E. R. (1988), 'Geometrical conditions for ballistic and visually controlled movements', *Ergonomics* **31**(5), 829–839.

Glenberg, A. (2007), 'Language and action: Creating sensible combinations of ideas', *The Oxford handbook of psycholinguistics* pp. 361–370.

Gold, K. (2009), 'An information pipeline model of human-robot interaction', *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction - HRI '09* pp. 85–92.

Goodrich, M. (2001), Experiments in adjustable autonomy, *in* 'Proceedings of IJCAI Workshop on Autonomy, Delegation and Control: Interacting with Intelligent Agents', pp. 1624–1629.

Graf, B., Hans, M. & Schraft, R. (2004), 'Care-O-bot II - Development of a next generation robotic home assistant', *Autonomous robots* **16**(2), 193–205.

Graham, P., Philippides, A. & Baddeley, B. (2010), 'Animal cognition: multi-modal interactions in ant learning', *Current Biology* **20**(15), R639–R640.

Grollman, D. H. & Jenkins, O. C. (2010), 'Incremental learning of subtasks from unsegmented demonstration', *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 261–266.

Grossman, T. & Balakrishnan, R. (2004), 'Pointing at trivariate targets in 3D environments', *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04* pp. 447–454.

Guglielmelli, E., Lauro, G., Chiarugi, F., Giachetti, G., Perrella, Y., Pisetta, A. & Scoglio, A. (2009), 'Self-feeding apparatus', *US Patent2009/0104004* .

Hancock, P. A., Jagacinski, R. J., Parasuraman, R., Wickens, C. D., Wilson, G. F. & Kaber, D. B. (2013), 'Human-automation interaction research past, present, and future', *Ergonomics in Design: The Quarterly of Human Factors Applications* **21**(2), 9–14.

Harnad, S. (1990), 'The symbol grounding problem', *Physica D: Nonlinear Phenomena* **42**(1), 335–346.

Harris, C. & Wolpert, D. (1998), 'Signal-dependent noise determines motor planning', *Nature* **394**(6695), 780–784.

Hartley, C. S., Cwynar, D. J., Garcia, K. D. & Schein, R. A. (1986), Capture of satellites having rotational motion, *in* 'Proceedings of the Human Factors Society 30th Annual Meeting', Vol. 30, pp. 875–879.

Hebb, D. O. (2005), *The organization of behavior: A neuropsychological theory*, Psychology Press.

Hochberg, L. R., Bacher, D., Jarosiewicz, B., Masse, N. Y., Simeral, J. D., Vogel, J., Haddadin, S., Liu, J., Cash, S. S., van der Smagt, P. & Donoghue, J. P. (2012), 'Reach and grasp by people with tetraplegia using a neurally controlled robotic arm', *Nature* **485**(7398), 372–375.

Holland, O. & Husbands, P. (2011), 'The origins of British cybernetics: the Ratio Club', *Kybernetes* **40**(1/2), 110–123.

Howard, A. & Roy, N. (2003), 'The robotics data set repository (radish)'.

Hsiao, K., Nangeroni, P., Huber, M., Saxena, A. & Ng, A. Y. (2009), Reactive grasping using optical proximity sensors, *in* '2009 IEEE International Conference on Robotics and Automation', IEEE, pp. 2098–2105.

Ijspeert, A. J. (2008), 'Central pattern generators for locomotion control in animals and robots: a review', *Neural Networks* **21**(4), 642–653.

Ingber, L. (1989), 'Very fast simulated re-annealing', *Mathematical and computer modelling* **12**(8), 967–973.

Ingram, J. N., Flanagan, J. R. & Wolpert, D. M. (2013), 'Context-dependent decay of motor memories during skill acquisition', *Current Biology* **23**(12), 1107–1112.

ISO (2000), Ergonomic requirements for office work with visual display terminals (vdts): Part 9. requirements for non-keyboard input devices (reference no. iso 9241-9:2000[e]), *in* 'Chapel Hill: University of North Carolina'.

Jardon, A. (2006), Assistive robot design methodology. Application to portable robot ASIBOT, PhD thesis, Universidad Carlos III de Madrid. Sobresaliente "Cum Laudem".

Jardon, A., Gimenez, A., Correal, R., Cabas, R., Martinez, S. & Balaguer, C. (2006), 'A portable light-weight climbing robot for personal assistance applications', *Industrial Robot: An International Journal* **33**(4), 303–307.

Jardon, A., Martinez, S., Gimenez, A. & Balaguer, C. (2008), Assistive robots dependability in domestic environment: the asibot kitchen test bed, *in* '6th IARP/IEEE-RAS/EURON Workshop on Technical Challenges for Dependable Robots in Human Environments', Pasadena, CA.

Kamrani, E. (2012), 'Real-Time Internet-Based Teleoperation', *Intelligent Control and Automation* **03**(04), 356–375.

Karwowski, W. (2012), 'A review of human factors challenges of complex adaptive systems: Discovering and understanding chaos in human performance', *Human Factors* **54**(6), 983–995.

Kelso, J. (1984), 'Phase transitions and critical behavior in human bimanual coordination', *American Journal of Physiology - Regulatory, Integrative and Comparative Physiology* **246**(2), 1000–1004.

Kim, D. (2012), 'How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot', *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* **42**(1), 2–14.

Kim, D., Sun, J., Oh, S. M., Rehg, J. & Bobick, A. (2006), 'Traversability classification using unsupervised on-line visual learning for outdoor robot navigation', *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* pp. 518–525.

Kim, J.-O. & Khosla, P. (1993), A formulation for task based design of robot manipulators, *in* 'Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on', Vol. 3, pp. 2310–2317.

Kline, R. (2011), 'Cybernetics, Automata Studies, and the Dartmouth Conference on Artificial Intelligence', *IEEE Annals of the History of Computing* **33**(4), 5–16.

Klyubin, A. S., Polani, D. & Nehaniv, C. L. (2007), 'Representations of space and time in the maximization of information flow in the perception-action loop', *Neural Computation* **19**(9), 2387–2432.

Klyubin, A. S., Polani, D. & Nehaniv, C. L. (2008), 'Keep your options open: an information-based driving principle for sensorimotor systems', *PloS one* **3**(12), e4018.

Knight, A. A. & Dagnall, P. R. (1967), 'Precision in movements', *Ergonomics* **10**(3), 321–30.

Kober, J., Mohler, B. & Peters, J. (2010), Imitation and Reinforcement Learning for Motor Primitives with Perceptual Coupling, *in* 'From motor learning to interaction learning in robots', Springer Berlin Heidelberg, pp. 209–225.

Kondraske, G. (1994), An angular motion fitt's law for human performance modeling and prediction, *in* 'Engineering in Medicine and Biology Society, 1994. Engineering Advances: New Opportunities for Biomedical Engineers. Proceedings of the 16th Annual International Conference of the IEEE', Vol. 1, pp. 307–308.

Kormushev, P., Calinon, S. & Caldwell, D. G. (2010), Robot Motor Skill Coordination with EM-based Reinforcement Learning, *in* 'Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on', IEEE, pp. 3232–3237.

Kremer, S. C. (2001), 'Spatiotemporal Connectionist Networks : A Taxonomy and Review', *Neural Computation* **13**(2), 249–306.

Lampe, A. & Chatila, R. (2006), Performance measure for the evaluation of mobile robot autonomy, *in* 'Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.', IEEE, pp. 4057–4062.

Lane, J. C., Carignan, C. R. & Akin, D. L. (2001), 'Advanced operator interface design for complex space telerobots', *Autonomous Robots* **11**(1), 49–58.

Liu, X., Mosier, K. M., Mussa-Ivaldi, F. a., Casadio, M. & Scheidt, R. a. (2011), 'Reorganization of finger coordination patterns during adaptation to rotation and scaling of a newly learned sensorimotor transformation', *Journal of neurophysiology* **105**(1), 454–473.

Lloyd, S. & Slotine, J.-J. (1996), 'Information theoretic tools for stable adaptation and learning', *Int. J. Adapt. Contr. Sig. Proc.* **10**(4-5), 499–530.

Louis, E. D. & Ferreira, J. J. (2010), 'How common is the most common adult movement disorder? update on the worldwide prevalence of essential tremor', *Movement Disorders* **25**(5), 534–541.

Lungarella, M., Pegors, T., Bulwinkle, D. & Sporns, O. (2005), 'Methods for quantifying the informational structure of sensory and motor data', *Neuroinformatics* **3**(3), 243–262.

MacKenzie, I. (1992), 'Fitts' law as a research and design tool in human-computer interaction', *Human-computer interaction* **7**(1), 91–139.

Mackenzie, I. S. (1989), 'A note on the information-theoretic basis for Fitts' law', *J. Mot. Behav.* **21**, 323–330.

MacKenzie, I. S. & Buxton, W. (1992), 'Extending Fitts' law to two-dimensional tasks', *Proceedings of the SIGCHI conference on Human factors in computing systems* pp. 219–226.

Mankoff, J., Fait, H. & Juang, R. (2005), 'Evaluating accessibility by simulating the experiences of users with vision or motor impairments', *IBM systems journal* **44**(3), 505–517.

Marocco, D., Cangelosi, A., Fischer, K. & Belpaeme, T. (2010), 'Grounding Action Words in the Sensorimotor Interaction with the World: Experiments with a Simulated iCub Humanoid Robot', *Frontiers in neurorobotics* **4**(May).

Mataric, M. J., Eriksson, J., Feil-seifer, D. J., Winstein, C. J. & Winstein, C. J. (2007), 'Socially assistive robotics for post-stroke rehabilitation', *Journal of Neuroengineering and Rehabilitation* **4**, 5.

McCrea, P. & Eng, J. (2005), 'Consequences of increased neuromotor noise for reaching movements in persons with stroke', *Exp. Brain Res.* **162**(1), 70–77.

McKinnon, M. & King, M. (1988), Manual control of telemanipulators, *in* 'Proceedings of the International Symposium on Teleoperation and Control', pp. 263–276.

Metta, G., Fitzpatrick, P. & Natale, L. (2006), 'Yarp: Yet another robot platform', *International Journal of Advanced Robotic Systems* **3**(1), 43–48.

Meyer, D. E., Abrams, R. a., Kornblum, S., Wright, C. E. & Smith, J. E. K. (1988), 'Optimality in human motor performance: ideal control of rapid aimed movements', *Psychological review* **95**(3), 340–70.

Mine, M. (1993), Characterization of end-to-end delays in head-mounted display systems (tech. rep. tr93-001), *in* 'Chapel Hill: University of North Carolina'.

Mitchell, J. & Shneiderman, B. (1989), 'Dynamic versus static menus: an exploratory comparison', *SIGCHI Bull.* **20**(4), 33–37.

Moller, M. & Polani, D. (2008), 'Common concepts in agent groups, symmetries, and conformity in a simple environment', *Artificial Life XI* .

Morse, A. (2010), 'Epigenetic robotics architecture (era)', *IEEE Transactions on Autonomous Mental Development* **2**(4), 325–339.

Nguyen, Q. & Kipp, M. (2014), Orientation matters: Efficiency of translation-rotation multitouch tasks, *in* 'Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems', CHI '14, ACM, New York, NY, USA, pp. 2013–2016.

O'Hara, J. M. (1987), Telerobotic control of a dexterous manipulator using master and six-dof hand controllers for space assembly and servicing tasks, *in* 'Proceedings of the Human Factors Society 31st Annual Meeting', Vol. 31, pp. 791–795.

Oja, E. (1982), 'Simplified neuron model as a principal component analyzer', *Journal of mathematical biology* **15**(3), 267–273.

O'Regan, J. K. & Noë, A. (2001), 'A sensorimotor account of vision and visual consciousness', *The Behavioral and brain sciences* **24**(5), 939–73; discussion 973–1031.

Pahwa, R. & Lyons, K. E. (2003), 'Essential tremor: differential diagnosis and current therapy', *The American Journal of Medicine* **115**(2), 134–142.

Park, J.-y., Chang, P.-h. & Kim, J.-o. (2006), 'A Global Optimal Approach for Robot Kinematics Design using the Grid Method', *International Journal of Control, Automation, and Systems* **4**(5), 575–591.

Park, J.-y., Chang, P.-h. & Yang, J.-y. (2003), 'Task-oriented design of robot kinematics using the Grid Method', *Advanced Robotics* **17**(9), 879–907.

Passenberg, C., Peer, A. & Buss, M. (2010), 'A survey of environment-, operator-, and task-adapted controllers for teleoperation systems', *Mechatronics* **20**(7), 787–801.

Pastor, P. & Righetti, L. (2011), 'Online movement adaptation based on previous sensor experiences', *IEEE International Conference on Intelligent Robots and Systems* pp. 365–371.

Patel, M., Miró, J. V. & Dissanayake, G. (2010), Dynamic bayesian networks for learning interactions between assistive robotic walker and human users., *in* R. Dillmann, J. Beyerer, U. D. Hanebeck & T. Schultz, eds, 'KI 2010: Advances in Artificial Intelligence', Vol. 6359 of *Lecture Notes in Computer Science*, Springer, pp. 333–340.

Pattacini, U. (2010), Modular Cartesian Controllers for Humanoid Robots: Design and Implementation on the iCub, PhD thesis, RBCS, Istituto Italiano di Tecnologia. Genova.

Paul, R. (1981), *Robot Manipulators: Mathematics, Programming, and Control - The Computer Control of Robot Manipulators*, The MIT Press, Cambridge, Massachusetts, USA.

Pavlov, I. (1927/1960), *Conditional Reflexes*, Dover Publications, New York, NY, USA.

Pelossof, R., Miller, A., Allen, P. & Jebara, T. (2004), 'An SVM learning approach to robotic grasping', *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on* pp. 3512–3518.

Pfeifer, R. & Bongard, J. (2006), *How the body shapes the way we think*, MIT Press, Cambridge, MA.

Pfeifer, R., Lungarella, M. & Iida, F. (2007), 'Self-organization, embodiment, and biologically inspired robotics', *Science (New York, N.Y.)* **318**(5853), 1088–93.

Pfeifer, R., Marques, H. & Iida, F. (2013), 'Soft robotics: the next generation of intelligent machines', *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence* pp. 5–11.

Pitzer, B., Styer, M., Bersch, C., DuHadway, C. & Becker, J. (2011), 'Towards perceptual shared autonomy for robotic mobile manipulation', *2011 IEEE International Conference on Robotics and Automation* pp. 6245–6251.

Prior, S. D. (1990), 'An electric wheelchair mounted robotic arm-a survey of potential users', *Journal of medical engineering & technology* **14**(4), 143–154.

Rocon, E., Gallego, J., Belda-Lois, J., Benito-León, J. & Pons, J. (2012), 'Biomechanical loading as an alternative treatment for tremor: A review of two approaches', *Tremor and Other Hyperkinetic Movements, http://tremorjournal.org/article/view/77* .

Rodriguez-Seda, E., Lee, D. & Spong, M. (2009), 'Experimental comparison study of control architectures for bilateral teleoperators', *IEEE Transactions on Robotics* **25**(6), 1304–1318.

Rosenfield, I. (1988), *The Invention of Memory - A New View of the Brain*, Basic Books, New York, NY, USA.

Rubchinsky, L. L., Kuznetsov, A. S., MD, V. L. W. & Sigvardt, K. A. (2007), 'Tremor', *Scholarpedia* **2**(10), 1379.

Sanger, T., Kaiser, J. & Placek, B. (2005), 'Reaching movements in childhood dystonia contain signal-dependent noise', *Journal of Child Neurology* **20**(6), 489–496.

Saridis, G. (1988), 'Entropy formulation of optimal and adaptive control', *IEEE Transactions on Automatic Control* **33**(8), 713–721.

Saridis, G. (1995), *Stochastic Processes, Estimation, and Control: the Entropy Approach*, John Wiley, New York, NY.

Sasaki, T. & Pratt, S. C. (2013), 'Ants learn to rely on more informative attributes during decision-making', *Biology Letters* **9**(6), 20130667.

Schaal, S. & Atkeson, C. (1998), 'Constructive incremental learning from only local information', *Neural computation* **10**(8), 2047–84.

Schaal, S., Ijspeert, A. & Billard, A. (2003), 'Computational approaches to motor learning by imitation', *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* **358**(1431), 537–547.

Searle, J. R. (2004), *Mind: a brief introduction*, Fundamentals of philosophy series, Oxford Univ. Press, New York, NY.

Sergio, L. E. & Scott, S. H. (1998), 'Hand and joint paths during reaching movements with and without vision', *Experimental brain research* **122**(2), 157–164.

Shannon, C. (1948), 'A mathematical theory of communication', *Bell System Technical Journal* **27**, 379–423.

Sharkey, N. & Sharkey, A. (2010), 'The crying shame of robot nannies: an ethical appraisal', *Interaction Studies* **11**(2), 161–190.

Shneiderman, B. & Plaisant, C. (2004), *Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th Edition)*, Pearson Addison Wesley.

Smits-Engelsman, B., Rameckers, E. & Duysens, J. (2007), 'Children with congenital spastic hemiplegia obey fitts' law in a visually guided tapping task', *Exp. Brain Res.* **177**(4), 431–439.

Smolensky, P. (1986), 'Information processing in dynamical systems: Foundations of harmony theory', *In Rumelhart, D. E., and McClelland, J. L (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition* **1**, 194–281.

Song, W. & Kim, J. (2009), Novel Assistive Robot for Self-Feeding, *in* A. Dutta, ed., 'Robotic Systems - Applications, Control and Programming', Vol. 1, InTech.

Soukoreff, R. & MacKenzie, I. (2004), 'Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI', *International Journal of Human-Computer Studies* **61**(6), 751–789.

Stolen, M., Dillow, B., Jacobs, S. & Akin, D. (2008), Interface for EVA Human-Machine Interaction, *in* '38th International Conference on Environmental Systems (ICES)', SAE, pp. 2008–01–1986.

Tani, J. (2003), 'Learning to generate articulated behavior through the bottom-up and the top-down interaction processes.', *Neural networks : the official journal of the International Neural Network Society* **16**(1), 11–23.

Tapus, A., Ţăpuş, C. & Matarić, M. J. (2008), 'User-robot personality matching and assistive robot behavior adaptation for post-stroke rehabilitation therapy', *Intelligent Service Robotics* **1**(2), 169–183.

Tikhanoff, V., Cangelosi, A., Fitzpatrick, P., Metta, G., Natale, L. & Nori, F. (2008), An open-source simulator for cognitive robotics research: the prototype of the icub humanoid robot simulator, *in* 'Proceedings of the 8th workshop on performance metrics for intelligent systems', ACM, pp. 57–61.

Tikhanoff, V., Cangelosi, A. & Metta, G. (2011), 'Integration of Speech and Action in Humanoid Robots: iCub Simulation Experiments', *IEEE Transactions on Autonomous Mental Development* **3**(1), 17–29.

Tonet, O., Marinelli, M., Citi, L., Rossini, P. M., Rossini, L., Megali, G. & Dario, P. (2008), 'Defining brain-machine interface applications by matching interface performance with device requirements', *Journal of neuroscience methods* **167**(1), 91–104.

Topping, M. (1993), Early experiences encountered in the placement of 60 handy 1 robotic aids to eating, *in* 'Proc. of the IEEE 93 Conference on Cybernetics', pp. 543–546.

Touchette, H. & Lloyd, S. (2004), 'Information-theoretic approach to the study of control systems', *Physica A: Statistical Mechanics and its Applications* **331**(1-2), 140–172.

Tsui, K. M., Feil-Seifer, D. J., Matarić, M. J. & Yanco, H. A. (2009), Performance evaluation methods for assistive robotic technology, *in* 'Performance Evaluation and Benchmarking of Intelligent Systems', Springer, pp. 41–66.

Tsui, K. & Yanco, H. (2009), Towards establishing clinical credibility for rehabilitation and assistive robots through experimental design, *in* 'Workshop on Good experimental methodology in robotics, Robotics Science and Systems', Seattle, WA, USA.

Tsui, K., Yanco, H., Kontak, D. & Beliveau, L. (2008), 'Development and evaluation of a flexible interface for a wheelchair mounted robotic arm', *Proceedings of the 3rd international conference on Human robot interaction - HRI '08* pp. 105–112.

Valavanis, K. & Saridis, G. (1988), 'Information-theoretic modeling of intelligent robotic systems', *IEEE Trans. Syst., Man, and Cybern.* **18**(6), 852–872.

Vanacker, G., Vanhooydonck, D., Demeester, E., Huntemann, a., Degeest, a., Brussel, H. & Nuttin, M. (2006), 'Adaptive filtering approach to improve wheelchair driving performance', *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication* pp. 527–532.

Vanhooydonck, D., Demeester, E., Hüntemann, A., Philips, J., Vanacker, G., Van Brussel, H. & Nuttin, M. (2010), 'Adaptable navigational assistance for intelligent wheelchairs by means of an implicit personalized user model', *Robotics and Autonomous Systems* **58**(8), 963–977.

Verschure, P. F. M. J., Voegtlin, T. & Douglas, R. J. (2003), 'Environmentally mediated synergy between perception and behaviour in mobile robots', *Nature* **425**(6958), 620–624.

Verschure, P., Kröse, B. & Pfeifer, R. (1992), 'Distributed adaptive control: The self-organization of structured behavior', *Robotics and Autonomous Systems* **9**(3), 181–196.

Waegeman, T., Antonelo, E., Wyffels, F. & Schrauwen, B. (2009), 'Modular reservoir computing networks for imitation learning of multiple robot behaviors', *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation - (CIRA)* pp. 27–32.

Weng, J., McClelland, J. & Pentland, A. (2001), 'Autonomous mental development by robots and animals', *Science* **291**(5504), 599–600.

Wenning, G. K., Kiechl, S., Seppi, K., Müller, J., Högl, B., Saletu, M., Rungger, G., Gasperi, A., Willeit, J. & Poewe, W. (2005), 'Prevalence of movement disorders in men and women aged 50-89 years (bruneck study cohort): a population-based study', *The Lancet Neurology* **4**(12), 815–820.

Wiener, N. (1965), *Cybernetics, Second Edition: or the Control and Communication in the Animal and the Machine*, The MIT Press.

Wiskott, L. & Sejnowski, T. J. (2002), 'Slow feature analysis: unsupervised learning of invariances', *Neural computation* **14**(4), 715–770.

Xiu, Z., Kitagawa, A., Tsukagoshi, H., Liu, C., Ido, M. & Wu, P. (2006), Internet-based Tele-rehabilitation System – Bilateral Tele-control with Variable Time Delay, *in* 'Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems', pp. 5208–5213.

Yamano, K., Tanaka, K., Hirayama, M., Kondo, E., Kimuro, Y. & Matsumoto, M. (2004), 'Self-localization of mobile robots with RFID system by using support vector machine', *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 3756–3761.

Yamashita, Y. & Tani, J. (2008), 'Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment', *PLoS computational biology* **4**(11), e1000220.

Yang, N., Jin, D., Zhang, M., Huang, C. & Wang, R. (2001), An extending fitts' law for human upper limb performance evaluation, *in* 'Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE', Vol. 2.

Yokoi, H., Arieta, A. H., Katoh, R., Yu, W., Watanabe, I. & Maruishi, M. (2004), Mutual adaptation in a prosthetics application, *in* 'Embodied Artificial Intelligence', Springer, pp. 146–159.

Yuste, R., MacLean, J. & Smith, J. (2005), 'The cortex as a central pattern generator', *Nature Reviews Neuroscience* **6**(6), 477–484.

Zahedi, K., Ay, N. & Der, R. (2010), 'Higher coordination with less control - a result of information maximization in the sensorimotor loop', *Adaptive Behavior* **18**(3-4), 338–355.

Zhai, S. (1998), 'User performance in relation to 3D input device design', *ACM SIGGRAPH Computer Graphics* **32**(4), 50–54.

Zhai, S. (2004), 'Characterizing computer input with Fitts' law parameters - The information and noninformation aspects of pointing', *International Journal of Human-Computer Studies* **61**(6), 791–809.

Zhai, S. & Milgram, P. (1998), Quantifying coordination in multiple DOF movement and its application to evaluating 6 DOF input devices, *in* 'Proceedings of the SIGCHI conference on Human factors in computing systems.', ACM Press/Addison-Wesley Publishing Co., pp. 320–327.

Zhong, J., Cangelosi, A. & Wermter, S. (2014), 'Toward a self-organizing pre-symbolic neural model representing sensorimotor primitives', *Frontiers in behavioral neuroscience* **8**(February).